CHAPTER 1

Introduction to Outsourcing

The topic of outsourcing has gained much attention in the press. A Gartner, Inc., report dated March 10, 2004 states, “The worldwide outsourcing market is expected to grow from $161.9 billion in 2002 to just over $235.6 billion in 2007 at a compound annual growth rate (CAGR) of 7.8 percent.” The article continues, “Because of the recent global economic downturn, cost reduction has been the primary driver for outsourcing over the past several years and continues as a strong driver even as economic growth returns.”[1] Outsourcing of IT services clearly will be a major venue for the fulfillment of our needs for IT systems for years to come. Figure 1-1 shows the breakdown for worldwide IT outsourcing and its respective growth rates.

[1] From a Gartner Dataquest Alert, “Forecast for IT Outsourcing Segments Shows Strong Growth,” by Bruce Caldwell, Rob De Souza, Allie Young, Ron Silliman, and Eric Goodness, dated March 10, 2004. Reused with permission from Gartner, Inc.

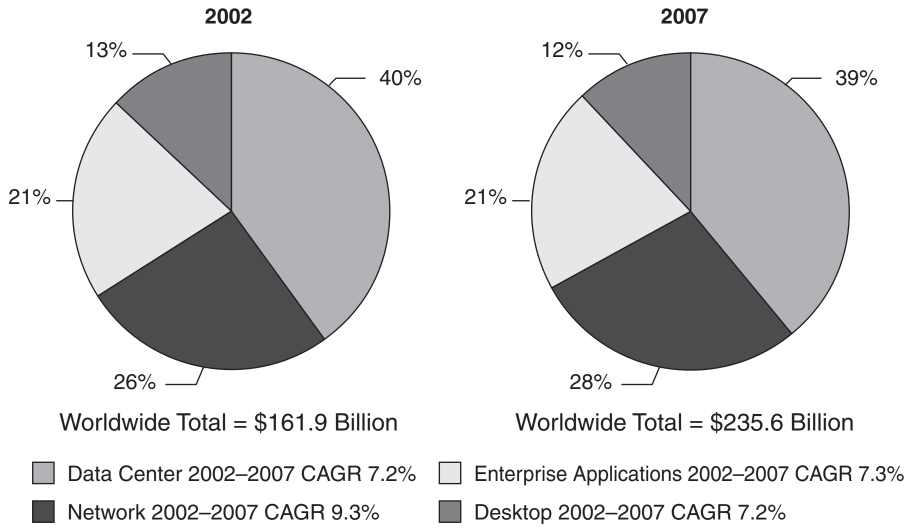

Figure 1-1 Worldwide IT Outsourcing Segments, Size, and Growth Rates, 2002–2007

From a Gartner Dataquest Alert, "Forecast for IT Outsourcing Segments Shows Strong Growth," by Bruce Caldwell, Rob De Souza, Allie Young, Ron Silliman and Eric Goodness, dated March 10, 2004. Reprinted with permission.

For a contractor, providing software development services to other companies is fascinating, challenging, and rewarding. It combines the technical challenge of applying technology to solve problems with the discipline of sound business management practices.

For a company contemplating a major software development project, outsourcing the work can be a convenient and cost-effective solution. This chapter sets the stage by describing the types of projects and environments frequently encountered in outsourcing situations.

Outsourcing Defined

Outsourcing is the practice of procuring IT services from sources external to an organization. It is important to distinguish between outsourcing and offshoring, which is a form of outsourcing. In recent years, the term "outsourcing" has become associated with organizations that employ overseas firms to perform the bulk of the work required by software development projects (offshoring), generally for cost-reduction reasons. Yet outsourcing (to domestic companies) has been used for many years, long before offshoring was commonplace.

This book defines outsourcing as "The process of retaining resources external to the procuring organization to conduct software development and related activities." The reason for this definition is as follows:

- It places no restrictions on the size or number of resources external to the procuring organization. The resources could be another company or an individual consultant.

- It places no restrictions on the location of the external resources in relation to the procuring organization; in other words, the resource can be located a long distance from the procuring organization or down the street.
- It places no restrictions on the number of organizations involved. This means that the procuring organization may hire a number of different companies or consultants, working either in a parallel or in a serial fashion. In other words, one contractor develops the system, and another contractor deploys and maintains the system after it is in production.
- It does not preclude the procuring organization from performing some of the work itself and outsourcing only a portion of a project to an external organization.

Notice that nothing in the definition specifically mentions overseas resources or offshoring. The practice of outsourcing to overseas companies for software development tasks is a fairly recent phenomenon. More traditional forms of outsourcing software development projects have been taking place for many years. In particular, the U.S. government has been outsourcing projects to domestic companies for dozens of years. That said, offshoring is increasing in popularity and also is covered in this book.

This is a broad definition of outsourcing. But it is necessary to include the multitude of scenarios that are commonly encountered today. The next section discusses the most common scenarios in outsourced projects, the variations within the scenarios, and the risks unique to each scenario.

Four Common Scenarios Encountered in Outsourced Projects

Software development projects present many challenges in an outsourced environment. In particular, these projects have the following challenges in common:

- Software developers are separated (sometimes by great distances) from the stakeholders.
- The software development team often has little knowledge of the problem domain.
- The stakeholders do not know how to express their needs in a way the development team can understand.
- The development team sometimes has difficulty articulating technical challenges to a nontechnical stakeholder group, who doesn't understand why something they are requesting may not be possible.
- Requirements frequently change.

As a result of these factors, communication between software development teams and project stakeholders is challenging at best. The result, in most cases, is wasted time and resources as development teams struggle to understand what to build and stakeholders struggle to use software that does not meet their needs.

The common thread in the challenges just listed is communication. Outsourced projects fall into four common scenarios (with some variations on each type). Each scenario presents its own advantages, disadvantages, challenges, and risks:

1. Projects in which the contractor is colocated physically with the outsourcing organization.
2. Projects that involve offshoring or employing a contractor's resources in another country, typically thousands of miles away. The work is performed at the contractor's location.
3. Variations on the first two categories, such as contractors separated a significant distance from the outsourcing organization, but still within the same country. In these situations, the work is done at the contractor's location.
4. Projects involving multiple subcontractors, typically seen in larger projects. In this scenario, the work is divided and spread across multiple contractors and the work is performed in multiple locations.

Scenario 1: Colocated Contractors

In this scenario, the contractor performs the software development activities in facilities provided by the outsourcing organization. Some companies prefer this scenario for outsourced projects, for the following reasons:

- The outsourcing organization may have security concerns over the business process or, more likely, the data involved in developing the system.
- The outsourcing organization believes that colocating the project will allow it to retain a higher level of control over the contractor and the project.
- The outsourcing organization believes that colocating project personnel makes it easier to communicate with the project stakeholders.
- The outsourcing organization can achieve some cost savings by providing facilities and IT services to the contractor.

At first glance, colocating the contractor with the outsourcing organization would seem to be a win-win situation for everyone. Yet, you must carefully consider some issues with this arrangement:

- Quality of facilities: Is enough space available? What about conference rooms and other auxiliary space?
- Will IT resources (PCs, servers, network) be provided? If so, how much control will the contracting team have over these resources?
- What are the facility's rules regarding working outside normal business hours?
- Problems could occur because of reliance on informal communication.

Let's examine each of these in detail.

Contractor Work Space Considerations

When considering whether to have contractors colocated in your facility, keep in mind that the contractor team will spend a good portion of their lives there. Their work space needs to be at least as spacious and comfortable as what they are accustomed to. Having a personal space that each team member can adopt as his own contributes to good team morale. In addition, the space needs to be as free of distractions as possible. Software engineers work on complex problems that require intense concentration. Numerous studies have shown the negative effects of frequent interruptions and distractions on productivity. Some project cost estimation tools, such as Cost Xpert Group's Cost Xpert, even accept as input a metric on the type of office space used for project personnel. Private offices are best, whereas open areas with no isolation between desks tend to have the worst effect on productivity. Don't forget to include several small conference rooms as well. Teams frequently need quick access to conference rooms to discuss possible solutions to problems. If these necessities are not available, allow the contractor team to work in their own facilities. Look to other areas and opportunities to reduce costs.

Quality of Facilities

This seems obvious, but poor facilities and working environments are a frequent complaint I have observed from colocated contractors. I have personally experienced situations of overcrowding—as many as two or three people placed in a cubicle designed for one. I have also seen poorly environmentally controlled areas (too hot, stuffy, odors) allocated to contractors. Finally, poor access to common areas (such as conference rooms with whiteboards, telephones that can make conference calls, and informal meeting areas such as kitchenettes) is a frequent complaint of colocated teams. Poor facilities are a major contributor to poor morale. The contractor will have to deal with complaints from team members, which is a distraction that results in turnover of otherwise-qualified team members. If this is a long-term project, or just one of many colocated projects, the story of the poor working conditions will spread rapidly through the contractor's company grapevine, which will cause the contractor to have difficulty staffing the project. An outsourcing organization that spends thousands of dollars per hour for a small- to medium-sized team and then forces them to accept poor or insufficient facilities will not receive good value from the team.

IT Resources

IT resources are another area that receives little attention proactively. When I mentioned that I would cover this topic in this book, one colleague was surprised. "What's the big deal?" he said. "You don't need to waste time on this. You just have someone do it." Of course, it's often not so straightforward. These are the sorts of problems and complications that occur:

- Developers often have PCs on their desks that are "locked down," meaning that they cannot alter the operating system settings or even install software without contacting their customer's IT department. They are not given administrator privileges on their own PCs.

- The PCs are often underpowered for development tasks. Developers tend to stress PCs and therefore require lots of CPU, memory, and disk space. Providing PCs that are generally used for processing e-mail and editing documents will not suffice for most development tasks. Because they do not own the PCs, developers often are not permitted to alter the hardware configuration, such as adding memory, even if they pay for the upgrades.
- The network connecting the PCs and servers is subject to the outsourcing organization's security rules and policies. Certainly the outsourcing organization has its own proprietary information on its network that needs to be protected, even from colocated contractors. The controls needed to protect these shared resources inhibit the contractor's ability to establish control over its own IT environment, which can make using certain tools difficult.
- Access to normal IT services, such as backing up and restoring files, can be more difficult, because the outsourcing organization's IT group may not be prepared to handle the needs of a software development team. Changes that normally are done in a matter of minutes at the contractor's home location can take days or even weeks at a customer site.
- The contractor is subject to changes and upgrades made on the network and IT environment, which may be scheduled with little notice and may occur at inopportune times. This is potentially disruptive to the development team.

IT Problems in a New Environment

I once worked as a consultant to a project for the U.S. Army. The development activities were to take place at the contractor's facilities. The contractor provided excellent hardware and infrastructure to establish a first-rate software development environment and associated tools for a multiyear project involving dozens of developers. A significant effort lasting several months was spent installing tools, configuring processes, preparing training, and customizing the tools to support and enforce the processes. Just as the software development environment was coming online, the contractor received word that the project was to be moved on-site with the customer. In addition, the contractor was not permitted to bring its PCs and servers. Everything had to be re-created on government-furnished equipment.

The transition to the new environment was difficult. The customer site IT environment had a policy against having any servers running Microsoft Windows, which prevented getting some of the development software operational. Although the desktop PCs ran Windows, the contractor had no domain administrator access. The on-site administrators gave all contractor requests very low priority, making it impossible to quickly transition to the new environment. To make matters worse, updates such as operating system patches and other upgrades were automatically pushed to all servers and desktop PCs without warning overnight. Developers would come into work some mornings and find that tools, or their application under development, would no longer work. Of course, this often happened at the worst possible times, such as right before a deadline.

Eventually, most of the issues were either solved or worked around, but in the meantime, hundreds of hours were lost. Developers could not wait for the tools to come online, so temporary methods and tools were devised to keep the team productive. This resulted in more work as the tools came online, because project data had to be migrated into the new tools.

Hours of Operation

Software development is not a 9-to-5 business. During peak project periods when the project is marching toward a deadline, overtime is common. When a bid for a project to be conducted on-site at a customer location is being considered, examine your company's history for past patterns of overtime on similar projects. Will the customer permit developers to work after hours and on weekends? If not, you need to plan for this time to be scheduled during the customer's normal business hours.

Overtime Issues

At one government site, it was known that the shop closed promptly at 7 P.M. and generally did not allow work to be conducted on weekends unless you were a government employee. Initially, this was not considered a problem. But as the project experienced some challenges (as most projects do), the overtime issue did become a problem. The customer indicated that on-site work could be conducted if a government employee accompanied the contractor. However, the request had to be made well in advance, which wasn't always possible. Furthermore, developers were reluctant to make the request because they knew it meant a government employee would have to take time out of her schedule to accompany the contractor during these overtime hours.

Communication Mechanisms

One of the advantages of being colocated on-site with the customer is quicker access to some of the stakeholders. This can also be a disadvantage. It is important to respect whatever formal processes are in place for requirements capture, as well as change requests to existing requirements. I prefer to caution stakeholders that change requests, bug reports, new requirements, and so on are to be submitted via a formal change control process, preferably whatever process the contractor defined. If they aren't submitted in this fashion, they risk being lost or forgotten. It's important to periodically remind your customer contacts of this.

Scenario 2: Offshore Projects

Projects that take place overseas face all the traditional challenges of software development projects and add many unique challenges:

- The contractor is physically located a long distance from the outsourcing organization and the project stakeholders. This raises a number of issues:
    - Real-time discussions, even via telephone, can be difficult, because the locations are a number of time zones away, and the work periods do not overlap or overlap only slightly.
    - E-mail threads discussing various topics with only a handful of replies may take a week or more to resolve. If these e-mails had stayed within the same time zone, they may have been resolved in only a few hours.

    - Face-to-face meetings are rare due to the expense and time involved with travel.
    - Network connectivity (particularly high-bandwidth connectivity) may be nonexistent or very expensive.
- The contractor is located in a different country than the country of the outsourcing organization. Many issues are related to this issue:
    - Differences in languages spoken.
    - Cultural differences.
    - Political differences. Some projects may have representatives from around the world, creating not only cultural differences, but also mixing nationalities who have historically been in conflict with each other. These situations can result in very politically charged situations.
    - Differences in laws protecting intellectual property.
    - Understanding that certain problem domains may differ significantly between the countries involved.
    - Differences in work practices in other countries.

Let's take a closer look at these characteristics.

Dealing with Offshore Projects

Successful projects usually result from building a high-performance team. The key to building a high-performance team is to establish trust through good working relationships with other members of the team. The distance between team members is a problem that can be partially overcome through the application of technology. Another contributor to building a high-performance team is to develop an understanding of the culture, laws, and business practices that are standard in the country where the other team members are located. When trying to build your connection with the offshore members of your team, you need to understand as much as possible about the cultural, work, and legal practices of the offshore country.

Using Technology to Help Establish and Maintain Communication

Although face-to-face communication is the most desirable form of communication, it's not possible for most offshore projects. Fortunately, technology offers many solutions that help alleviate these difficulties. Organizations considering offshore projects should think about adding the following capabilities for communicating with the project team:

- Audioconferencing equipment has greatly improved in the past few years. Despite being a relatively "low-tech" solution, audioconferencing remains an excellent low-cost solution for meetings. Recent innovations have reduced the "echo" that is common on overseas long-distance connections. This solution is best applied when most or all the conference participants are acquainted and can map the voices to the people on the team.

- Videoconferencing has been around for many years, but its use is still not very common. It is much less expensive than arranging face-to-face meetings with overseas participants, but it is often too costly for small companies unless its cost can be amortized over many projects. It also requires a large amount of network bandwidth, which may be prohibitive for connections to some countries. Still, it is worth investigating as an alternative, particularly for large companies in which the cost can be shared among projects.
- Electronic whiteboards are growing in popularity. Electronic whiteboards allow the writing on a whiteboard in one location to automatically appear in another location. They can be purchased for as little as $2,000 to $3,000. Installation and setup may involve additional costs, but the total cost is generally within reach of most small companies. When used in conjunction with a high-quality audioconferencing system, many of the advantages of videoconferencing can be obtained without incurring its cost.
- Many inexpensive group collaboration tools, such as scheduling, calendar tracking utilities, and address books, exist. Although they are relatively simple, I encourage usage of these, because they help you communicate major deadlines, meetings, and milestones across multiple teams.
- Many software development tools come with Web interfaces that can effectively be used over the Internet. The following are some examples from IBM Rational's product line:
    - RequisitePro provides requirements management capabilities. It offers a Web interface that contains most of the features of the native Windows interface.
    - ClearQuest is used for change request tracking. It is commonly used for defect and enhancement tracking, but it can easily be customized to handle nearly anything that is tracked via a formal process. This tool also offers a Web interface. An optional feature called ClearQuest Multisite is useful for geographically distributed projects in which high-speed connectivity is not possible.
    - ClearCase is a popular configuration management tool that lets you check in and check out files and manage baselines. This tool provides two features for remote usage. The Web interface is best suited for small projects. Like ClearQuest, ClearCase also has an optional feature called Multisite, which is highly recommended for geographically distributed development efforts.
    - Project Console is a unique product that extracts information from many other tools (mostly other IBM Rational tools) and produces metrics illustrating project progress.
- Project Web sites, or portals, are a useful means to exchange information. In the past, I have set up project Web sites for the exchange of project information. The project Web site can also function as a conduit for access to the Web-enabled development tools just described. Chapter 7, "Inception: Kicking Off the Project," contains suggestions on setting up a complete project Web site.

Understanding Differences in Cultures, Work Practices, and Laws

Some aspects to consider include the following:

- Some countries, particularly in Europe, have strict laws controlling or limiting the use of overtime. In those countries, it is a major issue to suggest that staff work extra hours. In other countries, such as Japan, long hours are common.
- Be careful about assuming basic knowledge surrounding the understanding of the problem domain. For example, if you are outsourcing a project to build a system for processing mortgages, does the country you are outsourcing to even use mortgages to acquire property? If not, you need to educate the contractor on the topic of mortgages (which wouldn't be necessary if the project were outsourced to a domestic contractor).
- Laws concerning intellectual property vary widely between countries. If the project being considered for offshoring involves sensitive information, be sure to consult an attorney who specializes in the intellectual property laws of the countries involved.
- Find out about the contractor's practices regarding trade-offs among schedule, quality, and completeness. In some cultures, releasing a product before it is fully completed or tested is unthinkable, even if this means missing a deadline.
- Understand the culture and practice of the contractor's approach to analysis and design. Some cultures tend to focus first on lower-level details, and others tend to focus on the big picture. Still other cultures insist on a full engineering approach and process.

Extended Vacations

I overheard one project manager discussing an interaction with his counterpart on a project that had been outsourced to a Scandinavian country. His company was unaware that it is often a custom in Scandinavian countries to take an extended vacation (often up to a month) in July, when the amount of daylight is near its peak. During that time, it was nearly impossible to get in touch with most of the key project members.

Further examples on this topic may be found in the publications listed in the bibliography. In particular, refer to Erran Carmel's excellent book, *Global Software Teams*.

Deciding Whether to Use Offshoring for a Project

By now, hopefully you understand that offshore projects can potentially incur a much higher amount of overhead, mostly due to the costs of establishing and maintaining effective communication. Although the labor costs may be much lower, you should consider this overhead and the delays inherent in an offshoring arrangement, as well as the additional risk incurred. It may be that these costs outweigh the potential savings involved. Yet, many successful offshoring projects have occurred. How do you decide whether to outsource to a domestic company or an overseas one? There is no magic answer, but you should consider the following points:

- Is the project's purpose to develop a new system? Will much requirements elicitation and business modeling occur? (See Chapter 9, "Navigating the Requirements Management Process.") For projects like these, extensive and frequent interaction with stakeholders and potential end users is necessary. Offshoring can be an excellent alternative if one or more of the following conditions are true:
    - The additional time and cost of enabling frequent contact (via travel) between the contractor and the stakeholders are not an issue.
    - The project is not on an unusually tight schedule and budget.
    - The contractor has built and successfully delivered many similar types of systems, and the teams that performed the work are available for your project.
    - Both the outsourcing organization and the contractor are well-equipped with the latest tools (such as audioconferencing and videoconferencing) to assist working in geographically dispersed environments.
- Is the project's purpose to augment or enhance an existing system? Offshoring can be a viable alternative in this situation. The added caveat (in addition to the previously mentioned criteria) is that the source code of the system being modified can be accessed, and so can the tools and documentation needed to build the system.
- Tasks involving testing and verification are excellent for offshoring. These tasks involve intensive "hands-on" work with a developed system and usually do not require much stakeholder interaction, except at the start and end of the task—ideal for an offshore environment. This is one of the few software development disciplines that can be sent entirely offshore. In most cases, with testing being one of the few exceptions, it is best to send entire subsystems or components offshore. The more robust and modularized your architecture, the greater the chances of success in offshoring certain components. The more dependencies there are between code developed in remote locations, the more difficult the task becomes.

Scenario 3: Distant Contractors, Same Country

This situation is similar to offshoring a project, except that the cultural issues are not as significant. In this case, the considerations for the type of project outsourced to a distant, same-country contractor are the same as for an offshore contractor. In addition, the same tools to enable communication between the outsourcing organization and the contractor are highly recommended.

Scenario 4: Multiple Contractors

In this scenario, the project being outsourced is large enough to divide the work among multiple contractors. The complexities in these situations can multiply quickly, depending on how the work is divided between contractors. Consider the following:

- Will the software being built by one of the contractors need to interface with the software being built by the other contractor?
- Will the contractors be working in parallel on different portions of the same system, or will they work on different tasks in a serial fashion?

If the answer to the first question is yes, or if the contractors are working in parallel, the following practices are recommended:

- During requirements elicitation meetings, all contractors should be represented. When partitioning the building of single system (or a "system of systems") is undertaken, the boundaries between the systems can be blurry. Because each contractor is focused on the performance of its individual contract, there is a natural tendency to avoid accepting responsibility for any work that is not clearly delineated as part of the contractor's Statement of Work. This can be a detriment to the system as a whole. When representatives from all contractors gather with representatives of the outsourcing organization, this is less likely to occur.
- When you form a Change Control Board, representatives from all contractors should be present. A change that seemingly affects only one part of the system could, in fact, affect the work of other contractors as well.
- It is incumbent upon the outsourcing organization to ensure that the vision for the system as a whole is consistent across all contractors. Is the system mission-critical? Is each contractor taking the necessary steps to ensure that redundancy, failover, and scalability are incorporated into the system architecture? Pay close attention to the supplementary requirements covering these areas.
- Will each contractor be developing portions of the user interface? If so, close coordination between the contractors will be necessary to ensure that the interface's look and feel is consistent. Also, each contractor should use the same technologies, languages, tools, and so on to build the system to make it easier to maintain the system after it is deployed.
- Do all portions of the system being built require persistent storage, or an application or Web server? If so, each contractor will want to build the system with its own servers, database instances, and so on. This is acceptable for the development environment, but if it is extended to the production operating environment, it will result in unnecessary duplication and increased costs. To avoid this problem, an appropriate arrangement must be made that consolidates these requirements without unnecessary duplication. At the same time, each contractor will want some control over its environment without delays. It may be appropriate for each contractor to have its own development environment, with the requirement that the production environment will run on a consolidated set of servers. Be sure that each contractor understands this.

- If one or all contractors are offshore, every contractor on the project needs to have the infrastructure to communicate effectively—not just between the outsourcing organization and each contractor, but between the subcontractors as well.

Overall, using multiple contractors increases the burden on the outsourcing organization. The outsourcing organization needs to have the staff to handle this added burden. This is discussed further in Chapter 4, "Best Practices for Staffing the Outsourcing Organization's Project Management Office (PMO)." In addition, a centralized repository for requirements, designs, documents, and so on is advisable. It should be accessible by all contractors involved in developing the system.

Where Does the Rational Unified Process Fit in All of This?

Thus far, we have discussed many of the challenges faced by outsourced projects, but we have not mentioned the Rational Unified Process (RUP). In fact, we have not discussed any aspects of software development. Why? The reason is simple. The challenges faced by outsourced projects mostly center on communication, cultural, business management, and other "soft" issues. A disciplined process (such as the RUP) does not provide all its intended benefits when the environment surrounding the project is in chaos or disorganization. Many projects that otherwise would have been successful have failed due to poor communication between the technical team and the business managers, poorly set expectations, and poor management from a business perspective. In these situations, no software process—RUP or otherwise—will result in success.

The RUP stresses close collaboration between teams and stakeholders. It is vital to break down as many communication barriers as possible before many of the RUP's benefits can be realized. The next chapter examines the RUP to help you understand what makes it different from traditional development processes, such as the Waterfall lifecycle process.

Summary

- When outsourcing to a colocated contractor, be sure to consider whether proper facilities can be provided. Consider allowing the contractor to create its own local-area network (LAN) that it can completely control.
- When working with a colocated contractor, be sure to adhere to any formal processes and mechanisms that are in place for managing requests, changes, and so on.
- During the beginning phases of a project that uses an offshore contractor, consider cultural awareness/sensitivity training to familiarize project staff with the business practices and philosophies common in the country in which the offshore contractor is located.
- If offshoring is to be used for a project that will require extensive requirements collection and analysis, be sure that the proper infrastructure is in place to provide the level of communication needed.

What’s Next?

The next chapter reviews and summarizes the Rational Unified Process for software development. I will compare and contrast it with traditional Waterfall lifecycle models. If you’re already familiar with the RUP, you might want to skim the next chapter or skip it and continue with Chapter 3, “Getting Started: Request for Proposals (RFPs), Proposals, and Contracts.”