

HOW OUTSOURCING AFFECTS TESTING

BY: REX BLACK

*This article is excerpted from Chapter 10 of Rex Black's upcoming book *Managing the Testing Process*, 3e.*

Over the last twenty years, outsource development of one or more key components in the system has come to dominate software and hardware systems engineering. The trend started in hardware in the 1990s. RBCS clients like Dell, Hitachi, Hewlett Packard, and other computer systems vendors took advantage of cheap yet educated labor overseas to compete effectively in an increasingly commoditized market.

Outsourcing spread slowly into software in the 1990s. However, in 2000, a series of events unfolded that accelerated the inevitable. The near-simultaneous bursting of the telecom and dot-com bubbles in 2000 combined with the Y2K and Euro-conversion wind-downs, making the software industry and its customers more conservative and price-conscious. This already-unsettled situation then collided with the ferocious 2001/2002 global recession. By the end of 2002, three years into a spectacular IT downturn that saw computer science enrollments in the United States fall to less than half of their 1999 levels, price had become the primary determinant in most IT project decisions. Mass outsourcing of software projects took hold, and it continues unabated to this day.

As with hardware, commoditization plays a role with software outsourcing. Software as a Service (SaaS) represents one facet of commoditized software, and is clearly a form of pay-as-you-go outsourcing. The use of open source packages represents another facet of commoditized software, being a form of pay-nothing-as-you-go outsourcing (albeit requiring potentially expensive self-support).

Another interesting factor to observe is that the Cold War ended in 1989, right as outsourcing as we know it today was born. The Cold War was a clash of ideologies between socialism and capitalism, and capitalism won. Capitalism is a system for maximizing return on capital, and one mechanism for doing that is reducing the cost of inputs required to produce an output. With the ongoing trends toward bundling applications with other applications and with computers, focusing on core competencies, commoditization of hardware and software, using cheaper offshore labor inputs to maximize return on capital, and insourcing of implementation of in-house IT systems, don't expect an end to this outsourcing trend any time soon.

I believe in the use of outsourced teams to provide key components and services as part of a project. If I didn't, RBCS would not provide outsource testing services. However, some RBCS clients have made the mistake of assuming that they could outsource without any risk, without any management or oversight, without any implications to their practices and processes. I have seen more than one manager, beguiled by clever marketing and sales presentations, impressive client lists, or low hourly rates, who then lost control of key risks to the project. Risks to system quality are among them. How do you manage testing when a constraint is outsourcing of some of the testing, all of the testing, and perhaps even the entire development effort?

In this article, I draw the scope of this discussion broadly, and examine the effects.

By outsourcing, I mean sending work outside of the organization, with variations that include the following:

- ◆ Outsource the development, but retain the testing in-house.
- ◆ Outsource the development and the testing to one company only.
- ◆ Outsource the development and the testing to two different companies.
- ◆ Outsource the development and/or the testing, each to multiple companies.

As you can see, this includes the distributed testing scenarios discussed earlier, but also includes other possible scenarios.

In addition, let's consider situations where the management of the project might be hands-on (directly, day-to-day direction of the outsourced work) or hands-off. Further, we have to understand that the location of the outsourcing might be across the hall in the same building, across street, across the town, across the country, or across the globe.

To manage testing in an outsourced situation, we also have to understand the reasons for it. These can include both spoken, readily acknowledged reasons, and unspoken, possibly present but often unacknowledged or even denied reasons.

The spoken, readily acknowledged reasons can include the following:

- ◆ The desire to realize labor and other cost savings.
- ◆ The expertise, capital equipment, or geographical advantages of outsource organization.
- ◆ The need for system or product certification by a qualified service provider.
- ◆ The inability to handle the work in-house, either due to a temporary spike in workload or a one-off project.

The unspoken, possibly present but often unacknowledged or even denied reasons can include the following:

- ◆ Organizational or peer pressure on decision-makers to outsource, which is the "everybody's doing it" argument.
- ◆ Dissatisfaction with the in-house team's capability, whether for service, cost, attitude, quality, or some other reason, which is the "it couldn't be worse, at least it's cheaper" argument.

Gartner Group calls outsourcing an "irreversible megatrend." So, love it or hate it, outsourcing is here to stay, for reasons spoken or unspoken, fair or foul. It's pretty much an oncoming train for us in this industry. As a test manager, your choices are stark and simple. You can get on the train and make it work. You can stand in front of the train and hope for the best. Or, you can change careers into a field you believe is not subject to outsourcing ¹.

I have come to the conclusion that the first choice is the only viable one for me and my company if we are to remain in the testing business. If you agree, let's look at how outsourcing affects testing.

INCREASED NEED FOR ORGANIZATION

To set the stage for properly utilizing outsourcing, the project will need to achieve a great degree of organization. Chaos will not scale across multiple organizations. This can pose a special challenge to organizations using Agile methodologies

like Scrum, since many of the enablers of these will not be present on outsource projects. You can't count on chance e-mails or hallway discussions to give the team information or clues about what is going well or poorly.

A number of our clients have not recognized that outsourcing greatly increases project complexity. Coordinating disparate groups of people, often separated by thousands of miles and many timezones, is a complicated endeavor. Like any complex human endeavor, careful planning, precise organization, and closely tracked and managed execution are key to success. This includes the need for careful mapping, integrating, and standardizing of the activities, as discussed earlier in this chapter.

I mentioned Agile methodologies a moment ago. I don't mean to suggest that you can't tailor these methodologies to work in an outsourced situation.

Whatever methodology your organization chooses, the chosen and possibly tailored development or maintenance lifecycle processes must have the following attributes:

- ◆ It must allow for distribution of work
- ◆ It must provide for checkpoints and course-correction at regular, predictable, reasonable intervals. By reasonable I mean not too frequent or too infrequent.
- ◆ It must include ways to reduce complexity in hand-offs, touchpoints, communications, change management, and so forth.

Unsuitable or cumbersome lifecycle processes can complicate, delay, or entirely block work on outsource projects.

For example, let's assume your project follows a sequential, V-model lifecycle. As shown in Figure 1, the project must have careful project management and change management activities in place at the end of each stage of the lifecycle. Otherwise, you run the risk of serious changes and deviations from plan accumulating in the project, which might only become visible during late-phase testing.

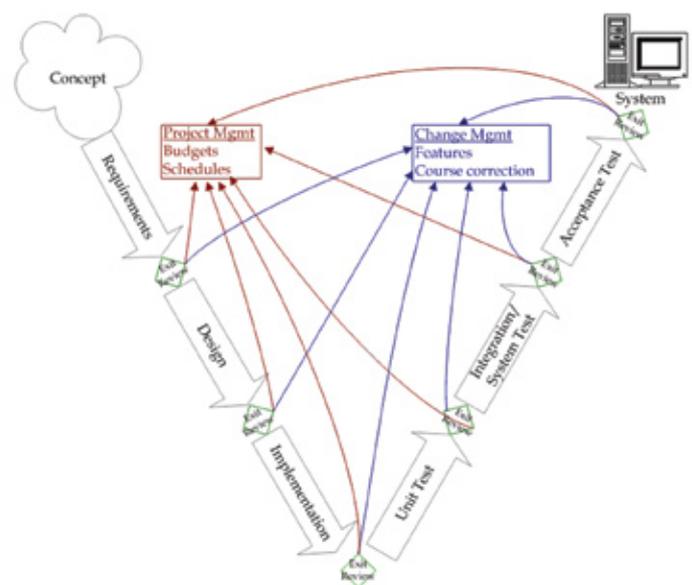


Figure 1: Project management and change management in outsourced projects

1 - For their full position on this, check www.gartner.com/press_releases/pr23june2003b.html.

SELECTING THE RIGHT TEST TEAM

Testing plays a key gate-keeping role in a project. When pervasive testing is practiced, as I have advocated throughout this book, it also provides an early-warning-system role. So, in an outsourced project, you have to select the right test teams to perform the various testing tasks, keeping both roles in mind.

If you intend to outsource some or all of the testing on a project, success depends on the following factors:

- ◆ You have to select the right testers for the testing tasks, with the right skills. This includes test management skills for large chunks of testing work that will be outsourced.
- ◆ The outsourced testers must have access to the right equipment, the right tools, the right infrastructure, and so forth. If the assumption is that the outsourced testing will use equipment, tools, and infrastructure that you have in-house, this will not happen by magic, but will require careful planning and management of the logistics of that.
- ◆ The outsourced testers must have the ability to adapt their work to your project and your organization.
- ◆ The outsourced testers must have sufficient independence to tell the straight truth about their results.

This all sounds obvious, I suppose, but, in my consulting work, I have seen two really risky outsource testing assumptions arise again and again. The first is that an organization can outsource both development and testing to a single organization, and that organization can (and will) test competently. The second is that, failing competent testing by the outsource development organization, acceptance testing by the contracting organization will suffice to prevent disastrous releases. As with any risks, you have to ask what the likelihood and impact will be if these assumptions are invalid?



REX BLACK

With a quarter-century of software and systems engineering experience, Rex Black is President of RBCS (www.rbc-us.com), a leader in software, hardware, and systems testing. For over a dozen years, RBCS has delivered services in consulting, outsourcing and training for software and hardware testing. Employing the industry's most experienced and recognized consultants, RBCS conducts

product testing, builds and improves testing groups and hires testing staff for hundreds of clients worldwide. Ranging from Fortune 20 companies to start-ups, RBCS clients save time and money through improved product development, decreased tech support calls, improved corporate reputation and more. As the leader of RBCS, Rex is the most prolific author practicing in the field of software testing today. His popular first book, *Managing the Testing Process*, has sold over 35,000 copies around the world, including Japanese, Chinese, and Indian releases. His five other books on testing, *Advanced Software Testing: Volume I*, *Advanced Software Testing: Volume II*, *Critical Testing Processes*, *Foundations of Software Testing*, and *Pragmatic Software Testing*, have also sold tens of thousands of copies, including Hebrew, Indian, Chinese, Japanese and Russian editions. He has written over twenty-five articles, presented hundreds of papers, workshops, and seminars, and given about thirty keynote speeches at conferences and events around the world. Rex is the past President of the International Software Testing Qualifications Board and the American Software Testing Qualifications Board.

Serious conflicts of interest occur when the developing organization tests its own work. Similar conflicts of interest occur if you insist on outside testing but allow the developing organization to hire the test partner. To list just a few of these problems, notice that independence of testing is compromised in both cases. Fresh perspectives might not always be available, even if the test partner is external (but hired by and working for the development organization). In both cases, financial incentives can work against you.

To address the second assumption, while acceptance testing of the deliverable is necessary, it is not sufficient. If you find serious problems during acceptance testing, what practical options might exist? Sure, your company could sue the development organization for failure to perform, but where does that leave your company in the interim during the lawsuit?

I suggest that, when outsourced development occurs, companies consider hiring an external testing services provider with no conflicts of interest. There are plenty of cheap, geographically convenient, and independent testing service providers available, no matter where you outsource the development. The modest increase in project cost and complexity is worth the peace of mind and reduced risk.

Another team-related challenge with outsourcing relates to the level of skills and, in many cases, the number of people at those skills levels, available in the outsource region compared to the outsourcing region. This is an issue because, in many cases, for reasons of labor pricing, outsourcing organizations choose to outsource to developing regions.

In a developing region, both the number of people with the appropriate level of skills at entry positions and the number of people with sufficient skills at senior technical and leadership positions might be limited compared to a developed region. You can see this by comparing Figure 2 with Figure 3. This is a hard matter to explore, for two reasons. First, the outsourced staff can only compare their skills relatively. Therefore, they will tend to measure their level of skills in relative terms, rather than against the level of skills available in the outsourcing region, which is often more developed. Second, the outsource organization has a strong financial incentive to inflate their skills relative to the skills available to the outsourcing organization.

This creates a real issue for testing in two ways. First, the outsource organization will tend to lack of deep skills and experience in fields that require them, such as performance testing, reliability testing, and test management. The problem is especially acute in terms of a lack of seasoned test managers, the kind of people who have been through dozens of projects and have a good ability to intuit and manage sources of test project risks. Second, because of the lack of deep skills and experience, outsource test teams in developing regions tend to lack the access to mentoring by highly skilled and experienced staff that people take for granted in developed regions.



Figure 2: Skills pyramids in developing regions



Figure 3: Skills pyramids in developed regions

Finally, since this topic comes up a lot in terms of selecting outsource development organizations, let's talk about what accreditation under CMMI or some other similar development process model does--and does not--tell you about testing and quality. To start with, let me say that I am not opposed to process formalization programs such as CMMI. I believe that they can result in quality improvement, when used properly. However, even Bill Curtis of the Software Engineering Institute admitted (at ASM/SM 2002 conference) that, when used purely as a marketing device, CMM does not result in significant quality or efficiency improvements².

Even if you find a vendor who is using CMMI properly and achieving quality and efficiency improvements from it, it's important to understand that CMM does not say much about testing. In other words, the various key process areas and key process indicators used in the CMM model are not the same as those one would use to assess testing capability. So, even an organization that achieves CMMI level 5 maturity for their software engineering processes can score low in test maturity.

To determine what we could and could not say about testing based on a maturity level rating, I did an analysis by reviewing the Capability Maturity Model against my own Critical Testing Processes framework³. The results are in Table 1. The twelve critical testing processes are shown in the first and third columns. The second and fourth columns show the corresponding level of CMM maturity that addresses that critical testing process. The numbers in italics indicate the CMM level that introduces some elements testing maturity to that critical testing process. The numbers in bold indicate the CMM level at which I think you could assume, with a reasonable degree of confidence, that the organization handled that critical testing process properly.

The bottom line is that, while CMMI and other similar models can help companies establish a foundation for quality, they do not guarantee thorough testing. When selecting the right test team, don't rely on remonstrations by the outsource development organization that, because they are CMMI level 4 or 5 or whatever, you don't need an outside testing process. For all the reasons covered in this section, while you might get away with that approach, it's at best a very risky one.

CRITICAL TESTING PROCESS	CMMT	CRITICAL TESTING PROCESS	CMM
TESTING OVERALL	2 3 4 5	TEST SYSTEM DEVELOPMENT	3
TEST TEAM ALIGNMENT	3	TEST RELEASES	2 3
RISK-BASED TESTING	3 4	TEST EXECUTION	3
TEST ESTIMATION	2 3 4	BUG REPORTING	3 4 5
TEST PLANNING	2 3	TEST RESULTS REPORTING	3 4
TEST TEAM STAFF AND SKILLS	3 5	CHANGE MANAGEMENT	2 3 4 5

Table 1: Measuring the testing maturity of CMM

2 - You can find an abstract of Curtis' talk at www.sqe.com/Events/archive/asm2002/key-notes.html.

3 - This test assessment and improvement framework appears in my book *Critical Testing Processes*. RBCS consultants have used this test assessment model to assess and improve test process capability for a number of clients around the world.

PLANNING AND PREPARING FOR TESTING IN OUTSOURCED PROJECTS

Okay, so you've selected an outsource development organization, hopefully also a testing service provider located fairly near the development organization who can provide truly independent testing during the development, and you've retained some portion of the final testing as an in-house effort. That's a good high-level view of how to organize the testing work on an outsourced project. However, it will not suffice for a plan. As with any distributed testing effort, testing in an outsourced project requires a good deal of planning.

Let's start with carving up the main testing activities. To do so, you'll need to refer to the lifecycle model that you are following. For the moment, let's assume you're following the V-model approach mentioned earlier. If so, then the division of work shown in Figure 4 makes sense. Combined with the project management and change management elements shown in Figure 1, you now have a model for knowing who is suppose to do what work when, and a way for measuring progress against major milestones and managing the inevitable changes that will occur.

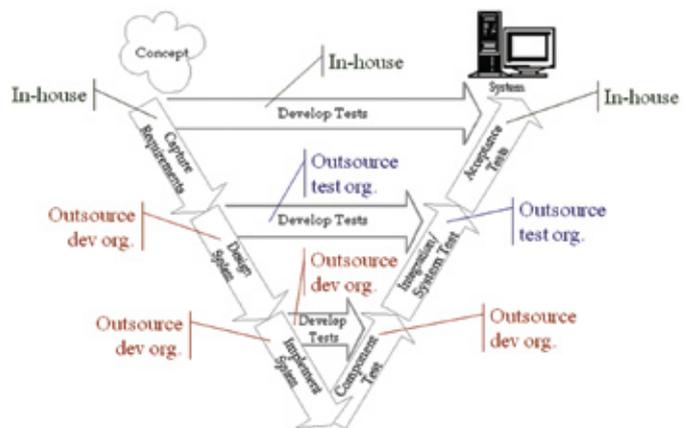


Figure 4: Carving up the test activities on an outsourced project

If the activities shown in Figure 4 were largely independent of each other, this might suffice. However, each activity generally produces a number of work products that must flow to other subsequent or concurrent activities. Further, these flows might actually be cycles, where a work product goes back and forth between various project participants, possibly spanning the organizational boundaries that exist. Therefore, you need to plan and ultimately execute crisp, efficient logistics for the testing activities.

While the specific logistical issues will vary from one project to another, there are three general categories that occur on just about any outsourced project:

- ◆ The system being built: This includes not only the transfer of test releases, release notes, code, and other supporting files, under proper configuration management control and as part of a solid release engineering process, but also the change management and project management elements.
- ◆ The test system: Due to reasons of expense or security, it is often necessary to share test environments with the outsource development and, if you have them, outsource testing partners. This includes the obvious elements of the test environment, but also access to cohabiting software, affected or linked systems or databases, and other connected elements that will be present in the customer

or production environment. You'll need also to be able to exchange various testware items, such as test data, cases, and test tools, both to audit the test work of the outsource organizations and to avoid problems with gaps and overlap in the creation of the testware. Finally, you also need to coordinate the test processes, at least major touchpoints that cross organizational boundaries.

- ◆ Information flows: A tremendous amount of information should flow across organization boundaries as part of testing in an outsourced project. This includes project documents, quality risk analysis documents, estimates and plans, work assignments, bug reports, and, of course, test results reports.

It's really easy to forget just one or two important logistical items. Doing so will cause immense headaches, often at a point in the project where it's become too late to resolve the logistical problem. Organizations running their first outsource project often have such logistical problems, but even experienced hands can get bitten.

On an Internet appliance project I worked on, we had a vendor who provided us with a modem integrated into the appliance. The modem consisted of both hardware and embedded firmware. During test execution, we started to experience severe unit-specific connection reliability problems. This vexed everyone, until one of the more hardware-savvy engineers compared the firmware from units without connection problems to the firmware from units with connection problems. Lo and behold, the vendor had changed the firmware without telling us. When we investigated the situation with the vendor, it turned out they had no configuration management, no source code repository worthy of the name, and no release engineering process. That one incident probably cost two person-months of project effort and could, had it happened closer to the project end date, have delayed release by up to a month.

Logistical risks are just one kind of project risk you need to manage on outsourced projects. Project risks on outsourced projects include all the usual ones⁴, plus a number of others such as the following:

- ◆ Political instability, especially when using outsource organizations in developing regions.
- ◆ Time zones, language barriers, and other communication issues.
- ◆ Infrastructure problems and inadequacies, such as unreliable Internet connectivity, poor roads and airports, difficulties obtaining potable water and safe food during site visits, and so forth.
- ◆ Skills availability, as mentioned in the earlier section.
- ◆ Unforeseen and sometimes abruptly exposed organizational weakness, including loss of key players due to extreme rates of turnover, organizational collapse due to governance problems, and the like.

Failure of the outsourcing organization to identify and manage these risks at a project level increases the potential consequences⁵.

One thing that can help you identify and manage risks, along with smoothing the way in which the work occurs, is to know the key participants in all parties. Ask yourself two questions. First, do you fully trust people you've never met? While you



4 - You can see Chapter 2 of my book *Managing the Testing Process* or Chapter 3 of my book *Advanced Software Testing* for a discussion of typical test-related project risks.

project with major trust issues, you know the answer to this question from having seen the results when trust is lacking. If you have to spend a lot of time and energy double-checking what people say and what they say they have done, that will reduce your efficiency, at best. At worst, trust issues lead to actual open or hidden enmity, possibly resulting in deliberate sabotage or undermining of other people's work and contributions in an attempt to prove a point or get even.

Trust is a major issue particularly with outsourced projects, because, frankly, the project team often starts out in a trust deficit. In many developed countries, there are reservoirs of fear and hostility among certain individual contributors and line managers who feel that their long term financial prospects, job security, and even ability to continue to work in their chosen career are threatened by outsourcing and insourced labor from offshore organizations (e.g., H1-B visa holders working for purely offshore staff augmentation firms). I believe most people will recognize that the individuals involved, both in the outsourcing organization and the outsource service providers' organizations, are simply relatively powerless players in the enormous geopolitical and economic re-ordering of the world that I mentioned earlier. However, recognizing that intellectually and dealing with the subconscious feelings associated with the creative destruction of capitalism in action are two different things.

Establishing trust in outsourced situation is also complicated by the fact that many times, the outsource service provider is a long, long way away. Most communication occurs through e-mail and conference calls. That can work for routine communication, but for tense, potentially fraught communications, these are problematic communication channels. You simply don't hear the whole message when you don't see the speaker's face, hands, and body.

I have frequently told clients that success distribution of testing, including outsourcing, requires a certain amount of jet fuel and a certain number of visas in a passport. Sometimes only a physical visit to the vendor's site will do. I believe this is especially true in three situations. First, when you want to evaluate the outsource company and their staff. Second, when you need to solve thorny project problems quickly. And, third, when you need to build and maintain trust and relationships.

This matter of trust is potentially complicated—but also potentially enriched—by cultural issues. Any time your testing involves an external partner, cultural issues become relevant, whether based in local culture or in the culture of a company. This is guaranteed by the fact that testers must hold different perspectives than other technical contributors.

My own background provides some examples. I changed from being a programmer and a system administrator to being a tester and test project leader when I took a job with a testing service provider. All my colleagues at the lab were testers. We worked on projects for clients who were developing hardware and software, but usually we interacted with only one or two client contacts. This all served to shield me from the political and cultural cross-currents that roil test groups in broader organizational settings.

I was forced to adapt rapidly a few years later when I became a test manager in a small custom software development shop. The political realities of testing within the context of a development organization are different, and I learned important lessons in that first test manager position. I have

5 - For example, at the time of writing this book, Indian outsource companies were reeling from a double whammy. First there was a spectacular terrorist attack in Mumbai. Then, just a few months later, one of the largest outsourcing vendors, Satyam, was revealed as an enormous (and almost bankrupt) piggy-bank for the owner and his family, complete with fake books and gulled auditors to hide the fraud. These two incidents illustrate the risk of political instability and that of abruptly exposed organizational weakness.

since relearned those same lessons, with a few variations, with subsequent employers and clients.

When you are involved in an outsourced project, you will experience these cultural issues writ large. Perspectives, priorities, and values differ from one team to the next even within a company, depending on the personalities of the team members, the leadership skills of the manager, the integrity of the perceived technical and moral leaders of the team (not just the manager), and, not least, the mission that team serves.

When you deal with external organizations, these cultural issues are intensified by the fact that the leaders of the partner companies can instill different values. Although I always emphasize the importance of individual contributors to the success of a team, I am often reminded, sometimes shockingly, of how much the vision, ethics, and leadership skills of a handful of top managers in a company can profoundly influence the way in which even the most mundane tasks are performed.

In terms of outsourced testing, such differences can mean that some individuals who would fail as test technicians, test engineers, or test managers in your company are seen as consummate test professionals perfectly aligned with the company culture in the outsource organizations. For example, I place great value on cooperative relationships between testers and developers. In contrast, however, some successful outsource organizations use an adversarial approach, with test managers encouraging testers to catch developers in errors. I would find it too emotionally draining to work in an organization where employees sought to undermine each other's successes, but if I worked with a test partner that used such an approach, it would hardly be within my purview to try to change that culture. Nevertheless, I might find the internecine quarrels distracting, to the extent that they spilled over into my life.

More subtle but equally challenging cultural clashes can occur. For example, I had a client who (at my recommendation) used a testing service provider in Taipei to handle some compatibility testing. My client's vendor was also located in Taipei. The testing service provider's corporate culture encouraged flexible but long hours, as my client did. The vendor involved had an 8-to-5 culture. Ironically, when we needed to institute regular conference calls, the vendor's culture and my client's culture meshed, but the time difference worked against the test lab. We had to schedule the calls for 8:30 a.m. Taipei time, which was fine for the vendor but troublesome for the project leader at the testing service provider, who worked from noon to midnight by choice. Ultimately, I decided to excuse him from the call, making sure I understood their results and could represent their findings. This allowed the test manager to continue working his comfortable schedule. Had I insisted that he start working cumbersome hours, that could have damaged our relationship.

Of course, there's also the matter of cultures in the sense of human culture. Cultures vary within countries, among religious and ethnic groups, and, of course, across countries. This is one of the really interesting and potentially fun parts of being involved on outsourced projects. You can use books like the Culture Shock series to gain insights into the local culture and how to work effectively with the people of that culture.

MAINTAINING FOCUS DURING TEST EXECUTION

If you've gotten yourself properly organized, as discussed so far, you'll have cleared most of the obstacles from your way prior to starting test execution. I certainly won't suggest that test execution will be trivial—it never is. However, beyond the

usual challenges of test execution, you have a few more to consider when dealing with testing on an outsourced project.

Most of these relate to the challenge of maintaining focus, including the following areas:

- ◆ Building the various project deliverables, testing them, gathering results, and reporting useful information to project stakeholders.
- ◆ Keeping communication channels open and clear, especially those channels that flow information into testing.
- ◆ Course correcting in response to triggered project risks and unforeseen problems.
- ◆ Managing change as the needs of the project and its stakeholders evolve.

It's important not to get distracted from your key testing objectives. Opportunities for distraction often abound on outsourced projects, including petty conflicts and turf battles, one-time crises that don't really offer any long-term lessons to learn, and minor differences in definitions, approaches, and other trivialities that don't affect anyone's ability to get work done. Remember that you can properly carry out all the stuff I mentioned earlier in this section, but, if you don't manage the test execution properly, you can end up not achieving anything of value.

CONCLUSIONS ABOUT OUTSOURCING AND TESTING

As for the writing of this article, the world economy, including software and systems engineering, had entered another tight economic period. Reducing cost is even more a by-word. Furthermore, while organizations still have a lot to learn about effective outsourcing of software development and maintenance compared to their hardware and systems engineering colleagues, many organizations have learned ways to make outsourcing succeed. So, outsourcing is here to stay.

As discussed in this section, outsourcing does pose a number of challenges to testing and quality. However, none of those challenges are fundamentally that much different or harder to manage than the risks and activities on non-outsourced, collocated projects. The differences are those of degree more than of kind. So, the diligent test manager can succeed on outsourced project, if she manages the overall testing process with the same level of attention to detail that she would use to manage any testing effort.

Indeed, I believe that, as software outsourcing matures, testers and test managers are well positioned to become essential project participants. Outsourcing might be cheaper than doing projects in-house, but it's harder, much more complex, and thus riskier. Since testing, ultimately, is a risk-management activities, I suggest you bring your risk-management skills to bear. You might well find that you are the key to software outsourcing success in your organization. ●

