An Outline for Software Testing Outsourcing[1]

Cem Kaner, J.D., Ph.D.[2]

Draft 6, revises the STAR draft

This document is intended as an issues list for planning and negotiating an outsourcing relationship. It is not intended as legal advice. The author makes no promise that this is suitable for your particular project or negotiation.

> "Outsourcing" refers to the process in which a company transfers all or part of one of its departments to an outside vendor who then handles the company's affairs for a price that is spelled out in the outsourcing contract. . . . When outsourcing occurs, the "customer" usually sells all or most of the related computer equipment it has on hand, either to an independent party or to the vendor (the outsourcer). Most of the employees who ran the equipment (i.e. the information systems (IS) employees) are either terminated or hired by the vendor. (Business Laws, Inc., "Introduction to Outsourcing", *Corporate Counsel's Guide to Outsourcing*, p. 1.001)

Companies outsource some or all of their software testing function for a variety of reasons—some good, some bad. Typically, they are looking for flexibility, cost control, access to equipment or specialized test materials, or relief from political infighting. Sometimes, they're looking for training, examples of good

---

[1] This is the final first draft of a chapter on Software Testing Outsourcing, to appear in Third Edition of Testing Computer Software. Comments and criticisms are greatly appreciated, and should be sent to kaner@kaner.com.

Several people have contributed to the development of this checklist. I particularly thank Hung Quoc Nguyen (LogiGear Inc.) and ST Labs for sharing source materials, Jim Bampos, James Bach, and Mike Powers for critiquing the work from the viewpoint of the outsourcer, George Hamblen, Frank Crowell, and Jeff Tatelman, for additional comments, and the Software Test Managers Roundtable (Chris Agruss, Jim Bampos, Sue Bartlett, Jennifer Brock, David Gelperin, Payson Hall, George Hamblen, Mark Harding, Elisabeth Hendrickson, Kathy Iberle, Herb Isenberg, Jim Kandler, Brian Lawrence, Fran McKain, Steve Tolman, and Jim Williams) for a detailed analysis of the outsourcing relationship in their first meeting, on October 31 and November 1, 1999 at the STAR Conference West, in San Jose, CA. Several of the notes here come from or were educated by that discussion.

[2] Cem Kaner is the senior author of TESTING COMPUTER SOFTWARE. He has worked with computers since 1976, doing and managing programming, user interface design, testing, and user documentation. Through his consulting firm, KANER.COM, he teaches courses on black box software testing and consults to software publishers on software testing, documentation, and development management. Kaner is also the co-founder and co-host of the Los Altos Workshop on Software Testing and the Software Test Managers' Round Table.

Kaner is also attorney whose practice is focused on the law of software quality. He is active (as an advocate for customers, authors, and small development shops) in several legislative drafting efforts involving software licensing, software quality regulation, and electronic commerce. He recently published a new book, BAD SOFTWARE: WHAT TO DO WHEN SOFTWARE FAILS (with David Pels. John Wiley & Sons, 1998).

Kaner holds a B.A. in Arts & Sciences (Math, Philosophy), a Ph.D. in Experimental Psychology (Human Perception & Performance: Psychophysics), and a J.D. (law degree). He is Certified in Quality Engineering by the American Society for Quality.

Kaner recently accepted an appointment as Professor of Software Engineering at Florida Institute of Technology and will take up residence there in August, 2000.

work, or an introduction to a new technology. And sometimes, the decision is based on an executive's whim.

This chapter is intended to help you think through the outsourcing relationship. Should you contract into it? What are the risks? The benefits? How do you structure it for success? How do you manage the inevitable problems and avoid (some of) the avoidable ones?

The core of the chapter is a long outline that identifies a few hundred issues in the test outsourcing relationship:

- When lawyers negotiate a contract, we often find it handy to work from an outline or detailed checklist of key issues. Fewer things slip through the cracks, and the negotiator doesn't have to rethink the same issues in every contract. Sometimes, the lawyers will even trade outlines. These can establish a shared list of issues, which can facilitate the haggling over the terms.
- Testers are used to outlines too. We use them to structure our test plans. Fewer things slip through the cracks. We might skip some parts of the outline, or test them lightly, but we do it with our eyes open.

Don't be Dazzled by the Outline

A key risk of this chapter is that it lays out so many details to think about (and potentially resolve) that it might push you into over-negotiating the deal.

A story often makes the rounds among lawyers about two companies (and their lawyers) who negotiated the perfect outsourcing agreement. After six months of intense negotiations, their contract covered every contingency. Unfortunately, a few weeks after the project finally started, the two companies had a disagreement. By this point, they'd used up their entire reserve of goodwill and all of their patience for negotiation. The project fell apart. All that negotiating served no purpose.

The lesson of this story is that you can't afford to negotiate every detail of a complex relationship. Instead, you want to reach agreement on the major issues, and on as many other points as you can comfortably cover. To handle the rest, you need a solid, friendly, fair process for making decisions and for resolving disagreements as the project goes forward.

In this chapter, I write from the perspective of the customer (the software publisher or developer who is contracting with a test lab), but I think the ideas are usually neutral. As an attorney, I've represented both, customers and labs. As a tester, I have retained labs and as a testing consultant I've helped labs work through the services they will offer customers. My goal is to provide ideas for managing the relationship and resolving disagreements, not to provide an advantage for either side.

Structure of this Chapter

This introductory material tries to explain the outline's structure. I cannot individually discuss each of the issues in the list, or at least, not without writing a book on the topic instead of a chapter. But we can look at some of the outline's complex or controversial issues.

The outline is divided into 7 key sections:

A. DECIDING THAT YOU WANT TO OUTSOURCE

B. FORMING THE RELATIONSHIP

C. ADDITIONAL STANDARD CONTRACTUAL ISSUES

D. MANAGING THE ONGOING RELATIONSHIP

E. TESTABILITY ISSUES

F. Other Standard Contracting Issues

G. Useful references

I'll discuss each of these briefly, pausing to add detail on some issues. I add detail:

- When the outline feels to me as though it has too much detail but the underlying concept isn't presented clearly enough,
- Or when I see a contractual issue that has caused parties to walk away from a deal (because they couldn't reach agreement) or has cost them excessive legal fees to work out. If I have something constructive to say (such as contract language that has seemed to address the issue, get the parties unstuck, and looks reasonable to me for both sides), I'll spend time on it.

Mylott (1995) presents and discusses many other contract clauses and legal issues in detail, as do several of the other references listed at the end of the chapter.

This chapter skips a significant class of issues, the human resources / employee relations / employment law issues. Several chapters in the *Corporate Counsel's Guide to Outsourcing* and in Mylott's *Computer Outsourcing* do a much better job than I can begin to do in this area.

Deciding that You Want to Outsource

This section presents benefits and risks associated with software test outsourcing. It asks what's in the deal for you, for the outsourcer, for other stakeholders in the company? Before getting into all of the details of the agreement, it's worth developing some confidence that a properly structured deal will actually work to your benefit.

As background for a decision like this, it's important to understand how test labs work, some of their constraints, and what they can provide.

Test labs typically sell services. You pay for time and expertise, with a limited additional cost for materials (such as equipment rental). Because the lab is selling you the time of its staff (or of contractors that it hires for your job), it has to mark up its costs. It's common for the lab to charge three to four times as much for labor as it pays. For example, if they pay $15 per hour to a staff member, you're likely to pay at least $50 for that staff member's time.

The better labs will try hard to give you good value for that extra cost, by training the tester and by supervising her. The lab's supervisory expenses are often invisible to you. There are often two types of supervisory expenses. You'll probably see (and pay for) the typical project management expenses. If you retain a test team, they'll typically come with a supervisor. But in addition to that, the lab will often take extra steps to make sure that the services provided are up to the standards of the lab. These are the supervisory services that are invisible to you. In the best labs, these are provided by the lab's top people.

A markup of three or four times the cost of the tester (pay $15, charge $45 to $60) might sound high, but there are many hidden costs. Along with the supervisory costs, there are the usual benefits-related expenses, training expenses, cost of advertising, cost of maintaining an up-to-date lab, and so on. My impression is that most or all of the labs who marked up services by only three times (pay $15, charge $45) have gone out of business. This is a tough business.

Test labs are under constant pressure to manage their labor costs. As a result, they can't pay most of their staff top dollar. Many labs have high staff turnover. Additionally, labs often go through feast and famine. When they take on your job, they might also be taking on several others that will compete for their time and their best staff. You cannot take it for granted that their facilities are any good, that their equipment is any good, that their staff are good or that you will get their best staff. You have to check them out, and you have to manage your relationship with them. The attentive wheel gets the good services.

In order to work well with the outsourcer, you will have to take on a communication burden. This is business as usual for some companies, but it might be very expensive for you. If you've relied on informal documentation (specs, test plans, etc.) up to now, you might find that the outsourcer needs more detail than you are used to providing or receiving. If you want high quality services from the outsourcer, you and your company might have to change practices, which might be an improvement or it might be a problem that is so serious for you that you can't afford to outsource.

Many of the better labs have gotten very skilled at dealing with last-minute receipt of products, without much specification or supporting test materials. They can do a very good job (within the usual constraints of time, money, and available hardware) of exploring your product, finding defects as they learn about the product, its market, its intended benefits, and its key weaknesses. But there are serious types of defects (which might result in data corruption, or unacceptable behavior when installed on important configurations, or intermittent crashes) that this style of testing can miss. And along with missing the serious but subtle bugs, the exploratory tester is not at all sure to cover all or even most of the code or of the features available to the customer. The exploratory tester works with what is visible, and especially on a rushed project, is likely to miss features that are more hidden.

If you have a reasonably experienced test group, the odds are that your staff are, on average, more experienced than the outsourcer's. Your staff probably understand your customers better and if you have a highly technical application, they understand the application better than the outsourcer. My best experiences in situations like this have involved bringing several of the outsourcer's staff onsite, to help the onsite staff cope with workloads that have gotten too high. Properly managed, the groups train each other.

If you are developing a product for the web or for another specialized environment, the outsourcer's staff might have *much* more experience and sophistication in that environment than you. The lab might have much better test equipment, and more experience using it, a better range of target systems (servers, clients, network setups, peripherals), and much more experience with compatibility testing. For specialized testing that takes advantage of genuine expertise of the lab staff, it makes more sense to do the work at the outsourcer's lab, no matter how experienced your staff. Unless you're retaining the outsourcer to help you set up your own configuration lab or put into place your own systems for using a critical new tool, you'll want the outsourcer's staff to work at the location where they are most effective (skilled at finding and troubleshooting problems) and efficient.

If you have a completely inexperienced test group, or no test group, you pose special challenges to a lab. Over the past 17 years, I've seen lab after lab start out by welcoming this segment of the market, then grow frustrated (and sometimes bitter), eventually deciding to look for customers elsewhere. One of the key problems is that the client company's representative is often a person who knows nothing about testing, who won't recognize good service or extra effort, and who negotiates primarily on the basis of time and price. In some cases, this person is openly offensive ("I need some monkeys to bang on my product"), without realizing it. This person also typically comes without much documentation, without anything else that will help the lab work more efficiently with the software under test, with a desired ship date that will be realistic only if his company isn't going to fix any of the bugs that the test lab will find, and without realizing what a heroic effort he is requesting. And he thinks that $50 per hour for the lab's staff is outrageously high. Some very good labs handle this situation well (often by selling fixed fee contracts—"we'll give you $5000 worth of exploratory testing") but some test groups that are less competent and less conscientious find this a hospitable niche.

If you are using an external lab to compensate for no in-house test group or a weak in-house test group, you should think long and carefully about whether you should instead set up and fund a good group of your own. The outsourcing company is no silver bullet but even if it was, silver bullets hurt as much as lead bullets when you shoot them into your own foot. If, after thinking through your alternatives, you

decide that you prefer to use an external lab or that you have no realistic options, you should still plan to play an active role in supervising and understanding the relationship and the testing.

If you don't have a management strategy, you are less likely to achieve the benefits of outsourcing.

James Bach's papers on outsourcing were written while he was Chief Scientist at ST Labs, which was then the largest software test lab in the country. They are very thoughtful. If you can no longer find them at www.stlabs.com, you can get them from him directly. Check his website, www.satisfice.com, or contact him at james@satisfice.com.

Forming the Relationship

Once you've decided that outsourcing is worth pursuing, you have to find the right outsourcer and work out key details, such as schedule, price, and services to be provided. I think that two other fundamental details, to be taken care of up front, are the dispute resolution procedures and the intellectual property rights. The issues raised in this section will have day-to-day implications for the work of the lab and of your test group. The issues in the next section also have ongoing impact, but they are more visible to attorneys than engineers.

Qualifying the Lab: The Problem of "Professionalism"

In several consulting services contracts, there is a boilerplate promise (warranty or representation) to the effect that the consultant will provide services that meet the highest professional standards or that are consistent with normal professional standards in the field.

Up to this point, it has been difficult to hold software developers and consultants liable for malpractice (Kaner, 1996a). (There have been several malpractice-related settlements as part of the Year 2000 mess, but I'm not aware of a shift in the published case law in the United States since 1996.)

However, the rules change once the state licenses a profession. At that point, the members of the profession are required to exercise the diligence and skill and knowledge that one would expect from a reasonably prudent, competent member of the profession. There has been a big push recently to "professionalize" the field of software engineering, by licensing software engineers as professional engineers. At time of writing, Texas, British Columbia, and Ontario license software engineers. For a detailed review, see McConnell & Tripp (1999) and other articles in the same issue of *IEEE Software.*

As I understand Texas law, independent contractors and consultants who engage in software engineering can be held liable for malpractice if they fail to meet the appropriate level of care (diligence, skill, knowledge, etc.).

My core objection to licensing is that I don't believe that we have achieved consensus on what the appropriate standard is or should be. Less than a year ago, I was confronted in public, at a conference by a screaming consultant who called me a liar. As far as I can tell, his primary objection was that we disagree on the definition of "black box testing." Even the simplest basics, like that definition, are subject to vehement disagreement among senior members of the profession.

As an attorney, I am confident that I can conduct my legal practice in a way that will serve my clients well, without exposing me to risk of losing a malpractice suit.

As a software engineer, I have no such confidence. What seems to be good practice to me might seem to be bad practice to some of my peers (and vice versa).

As an attorney who represents software engineers, I think that a professional licensing system at this point will create a strange lottery system. If someone is disappointed with your work, and they sue you for malpractice, the reasonable standard of professional care and conduct will be invented, case by case, by insurance companies and jurors. I have no confidence that this will be fair to the parties in the lawsuit or that it will lead to sensible maturation of the field.

Because of the increased (to my eyes, random) liability risk, I expect to see outsourcers proposing clauses like the following in their contracts:

> *The services of CONSULTANT shall not be construed as "software engineering" services. CONSULTANT has not represented that he is a software engineer, nor that he provides software engineering services. CONSULTANT does not promise, and CLIENT does not expect, that the services provided under this contract will conform to whatever standards might be applicable to services offered by individuals who are licensed as professional software engineers.*

As a software testing consultant, I will not contract to provide my services in Texas unless the contract has this clause or one like it.

Certification

The client wants you to certify that the software is of high quality, and it will advertise your certification to other people (customers or shareholders).

In *Hanberry v. Hearst Corp.* (1969), the publisher of *Good Housekeeping* magazine (Hearst) was held liable for an injury caused by a defective product because it had given the product its "Good Housekeeping's Consumer's Guaranty Seal." *Hempstead v. General Fire Extinguisher Corporation* (1967) involved allegedly negligent certification of the safety of a fire extinguisher by Underwriters Laboratory. In *FNS Mortgage Service Corp. v. Pacific General Group, Inc. and International Assoc. of Plumbing and Mechanical Officials* (IAPMO) (1994), the court allowed a suit to proceed against IAPMO for negligent certification. In each case, the plaintiff (the person bringing the lawsuit) was a third party—a user of the product rather than the manufacturer.

If your certificate of quality or standards compliance is relied on by users, and if the product injures someone, costs them money, or destroys their business because of its defects or noncompliance, you are in trouble. Think carefully about whether you want to be a certifier.

No Third Party Reliance

The *Hempstead* case illustrates an important point. People bought the fire extinguishers because they knew that the extinguishers had been tested by Underwriters Lab. Customers therefore argued that they should be able to hold UL liable, even though its contract was with General Fire Extinguisher, not with them. In general, it might help you fight off a claim that you owe a duty to a third party if you include the following language in your contract:

> *<u>No Third Party Beneficiaries</u>. Consultant provides the Services solely for the benefit of the Client. As the sole intended beneficiary, only the Client has the right to enforce this Agreement. This Agreement is not enforceable by any third parties, including, without limitation, the Client's customers, creditors, potential investors, shareholders, or other developers or testers who are working for, or in conjunction with the Client or with Consultant.*

For more discussion of third party issues that led me to draft this clause, see the American Law Institute's *Restatement (Second) of Contracts*, sections 302, 304, and 315.

I also like the following language:

Class of Persons to be Guided by Information Supplied by Consultant: *Information that Consultant provides to Client is for the benefit and guidance of Client only and is not intended for communication to a broader audience. Client will not advertise or publicize Consultant's role in the testing of this product in any way that is calculated to increase the confidence of potential customers, investors, or other third parties in the reliability, usefulness, or value of this product.*

For more discussion of the issues that led me to draft this clause, see the American Law Institute's *Restatement (Second) of Torts*, Section 552.

If you're exceptionally successful as a consultant, you might include another clause, requiring the client to indemnify you (pay all your expenses and losses) in the event that you are sued by a third party, such as an end customer of the client. This is a logical clause to include, but indemnification clauses will catch the attention of the client's lawyer. If you ask for too much in your contract (or in your revisions to the client's contract), you risk losing the deal altogether or getting a deal that carries a lot of baggage from an adversarial negotiation.

No Life-Critical Applications

Your methods might or might not be appropriate for life-critical or mission-critical applications. Your schedule might or might not be appropriate for critical applications. If you intend to do critical applications, ignore this section. If you don't intend to do critical applications, make that understanding explicit in your contract. If it turns out that someone uses the tested software for life-critical tasks, and that software fails, this clause should help you argue that you're not the right person to confront over the failure of the software to meet the very high standards appropriate for critical software:

Consultant's Testing is Insufficient for Life-Critical or Mission-Critical Software: *Consultant helps its clients search for problems that might make a product unsuitable for normal commercial use. Consultant strives to be particularly helpful for clients who are operating under challenging deadlines. Consultant does not provide, and Client is not requesting, exhaustive analysis and testing of the product.*

Consultant explicitly cautions Client that Consultant's methods are not sufficient for testing a product for potential failures that could threaten human safety or the economic survival of a business. Consultant's systems-level approach might be very beneficial as part of a larger testing effort for such a product, but Client is advised and understands that Consultant does not claim to provide, and does not claim to be competent to *provide*, the additional work that would be necessary to assure that this product is safe or that it is reliable to a known and quantifiable degree of reliability.

Dispute Resolution

The typical contract either leaves disputes to the courts or includes an arbitration clause. An arbitration is a formal proceeding, like a trial, but it's run privately rather than by the government. The problems in both cases are essentially the same. By the time you get this far, you and the test lab will probably be furious with each other. The project will have fallen apart long ago. All that's left to argue about is who was right, who was wrong, who owes who money, and how much. This is a disaster, not a dispute resolution process (Kubey, 1991).

Some day, you might need to resort to trial or arbitration after a project fails. But this shouldn't be the only dispute resolution process, or the main one, in your contract. You should make it as easy and as natural as possible to identify and resolve problems quickly, long before they get this serious.

The goal of a dispute resolution strategy should be to help you finish the project successfully, while preserving your good relationship with the test lab. Anything else is a project failure.

The Liaison

The foundation of effective dispute resolution is good communication.

Your best strategy for preventing, detecting, and quickly dealing with problems with a test lab is to appoint someone from your staff to work on the project with the test lab.

This is not a controversial view:

- Mylott (1995) calls this person the *liaison*.
- XXCAL asks its customers to make sure that "a client representative be available for telephone support during the entire course of the test cycle."(XXCAL, 1997)
- ST Labs used to urge its customers to appoint an internal test team to delegate work to the lab and supervise the results on an ongoing basis. "Over the course of hundreds of projects, we've found that an outsource project probably will not satisfy either party if there isn't frequent communication between them."(Bach, 1997; Kaner & Bach, 1995)

The liaison serves as the primary contact between your company and the test lab. She is the primary reviewer of the lab's work, including test plans, bug reports, status reports, test suites, and all other deliverables from the lab to you. She also works to understand this lab's business and technical practices and the practices of test labs in general. With this knowledge, she provides your company with insight and she serves as a more effective negotiator with the lab.

The liaison is the person that your staff complain to when they have problems with the test lab, and she is the person that the lab staff complains to about you. For example, when one side says that the bug reports are badly written and irreproducible, and the other side says that the programmers are just trying to avoid taking responsibility for their own bugs, your liaison is the person who is digging through the bug reports trying to understand who needs more technical training and who needs attitude adjustment.

Your liaison should be calm, able to assert herself without shouting, and able to deal with pressure and whining from all sides. She should be diplomatic but firm, able to explain your company's needs to the outsourcer without being obnoxious, and able to present the outsourcer's legitimate gripes with your company to your management without getting fired. She should be an excellent listener, tightlipped, and personable enough that the outsourcer will confide in her. She should have, and deserve, an air of solid credibility and integrity. She should be detail oriented and methodical. She should understand testing, test management, software development, and software project economics. Finally, she must be loyal to your company.

The test lab will probably have its own liaison (they might call him an account manager or a project manager).

The two liaisons, yours and the lab's, are the first-level negotiating team for resolving problems as they arise.

Disagreements

What kind of disagreements come up between a software developer and the test lab? Here are some examples:

- The latest version of the software is too unstable to test. It takes days (or weeks) for the developer to deliver a stabilized version. What is the test lab's staff supposed to do during this time? What if they have little productive work to do on your project? Should they work inefficiently, at your expense? Should the lab reassign them to another project, saving you money but possibly imposing an additional delay until they can rejoin your project?
- The lab promised four seasoned testers, full time. It seems to you that you're only getting about two testers' worth of productivity. Testing is going very slowly, but the lab has no additional testers to add to the project.
- The lab is testing your competitor's software at the same time as yours. The same testers are seeing both products. One tester told your liaison about a slick new feature in your competitor's program. You believe that your competitor would consider this feature a secret. You don't want your competitor to learn about your slick new features.
- You loaned the lab several computers and printers. It has returned a few to you, but they were so badly packaged that they were trashed in shipping. The lab also seems to have lost your $10,000 color laser printer, and they're asking you about some equipment that they claimed they loaned you, that you've never heard of.
- The lab gave you a sketchy test plan and you didn't like it. You made extensive comments. They didn't make many changes, and you're still not satisfied. The lab is entitled to a payment when they reach the first milestone, "Test Plan Draft 1 Complete." They want their money, but you want a better first draft.
- The lab's testers seem clueless, at least the ones assigned to your project. The people that you met when you interviewed the lab, and toured the lab, were very sharp but none of them are assigned to your project. You want testers who are more senior, but the lab says they'll cost more money. You don't want to pay more because you thought that the rate you'd agreed to was for the senior testers.
- The lab agreed to test your software for a flat rate of $100,000. Since then, you've changed the feature set six times. The latest feature includes modem support, and now several people can use the program at the same time, seeing and editing each other's data. You also want the program tested for compatibility with the new (still in beta) Windows 98 and NT (these didn't used to be a requirement, but they are now). The test lab wants more money to pay for the extra testing. You want to pay the flat fee, $100,000.

There are plenty of other possible problems, but these illustrate the point. Some of these are big—a lot of money is involved. Others might just involve some retraining and some resetting of expectations.

A Strategy for Resolving Disagreements

You want to identify and solve problems as early as possible. The longer they drag on, the more damage they do. They hurt the project, and they hurt your ability to trust and work with each other. Mylott (1995) also discusses a step-by-step gradual dispute resolution in detail and he suggests contract language.

- **Start with discussions at the liaison level**

  If the liaisons can straighten the problem out, no one else has to deal with it.

Negotiating takes some skill and your liaison may need training in it. You want to use ethical methods of negotiating with the goal of preserving a business relationship. (Freund, 1992; Fisher & Ury, 1991).

- **Gradually escalate through management levels**

  If the liaisons can't resolve the disagreement, they should both take up the matter with their managers. The contract should describe this as part of the process—the first level managers will "meet and confer" to try to resolve the problem. If they can't reach agreement, the dispute goes up another level, and two more managers are required to meet and confer.

  Sometimes, this sounds better than it works. Go up a few management levels and neither side might understand the issues or the technology. The result can be a deadlock in negotiations or a set of uninformed and ill-advised decisions.

- **Use independent experts for fact-finding, arbitration, and mediation**

  Rather than continuing up the management chain, it might make sense to try something else. If your companies can't reach agreement when the first or second or third level (decide what level and write it into your contract) managers meet, get some help from an independent expert.

  The expert should be someone that both of you trust. In your contract, you should list a few mutually acceptable people. You want to make this list before there's a problem because it can be difficult to agree on which people to consult if you wait until you're in the middle of a significant dispute. The list includes a few people so that if one isn't available, you have other choices.

  The expert should have no other role in the project or in either of your companies—you want to avoid conflicts of interest. When you have a dispute to settle, and you call one of the experts for help, require him to list his history with both sides and any other conflicts of interest involving the project. If there are conflicts, you should be able to refuse the services of this expert. Call the next person on the list.

- **Independent fact-finding.**

  In this case, the expert meets with your staff and the test lab's staff. He looks at documents and writes a detailed report that describes the disagreement between you and the lab. He recommends a course of action. The report should be direct, but should not insult either side. The report goes to the managers who couldn't reach an agreement before. They should then meet and discuss it.

  If the main problem was a misunderstanding, or if one side was trying to get away with something that just won't fly once the other side understands what's going on, the expert's report should help settle the dispute.

  Even if you and the lab don't reach an agreement when you read and talk about the report, the report will provide useful background to the mediator or the arbitrator.

- **Mid-project mediation.**

  A mediator doesn't have to be an expert in the technical matters because he doesn't make the decisions. But he must be literate in the technology. The mediator meets with you and with the lab, tries to understand each of your positions, tries to help each of you

understand the other's point of view, and tries to help you do creative problem-solving. The goal is to help you both reach a agreement that satisfies both of your needs. Often the agreement that results from mediation involves a plan that neither side would have thought up on their own.

If you reach an agreement, write it down, sign it, and move forward with the project. If not, try arbitration.

- **Mid-project arbitration.**

  This is a session lasting 4-8 hours, in which both sides present the dispute to the expert. The arbitrator listens carefully, asks questions, and issues a ruling within a week. She decides who has to do what and who has to pay for it. The goal is to get a clear decision on the issue right away, and move forward. Even if the arbitrator makes the wrong decision, it is often better to move forward than to drag down the project by prolonged fighting over an issue that won't go away.

  The contract should let you and the lab settle the dispute on your own terms before the arbitrator gives her decision. People often settle disputes at the last minute, when it looks like someone else will make the decision for them if they don't deal with their issue themselves. If you reach an agreement after the hearing and before the arbitrator gives a decision, call the arbitrator and tell her not to give you a decision.

  Your contract might have special rules for large disputes (involving large sums of money). For example, the negotiations might go to more senior managers before they go to arbitration. You might use a three-arbitrator panel instead of a single arbitrator. At least one of the arbitrators should be a lawyer with extensive arbitration experience. The hearing might last longer than a day. The goal is still the same—make a decision quickly and get the project moving again.

Conclusion (Dispute Resolution)

A contract should provide rules for protecting and preserving the deal you've agreed to, not just rules of engagement when the deal falls apart. It should help you protect the deal by providing simple rules for settling disagreements before they kill the project. The approaches discussed here are designed to help you resolve disagreements while your project is still alive and promising. They appear too rarely in contracts. Think seriously about including them in yours.

Intellectual Property Issues

Several intellectual property issues must be sorted out. We'll look shortly at infringement-related liability. Here, I raise the issue of ownership of the outsourcer's work product.

The standing assumption in the Copyright Act is that if Person X wrote something, Person X owns the copyright on it. If Person X is the outsourcer, and what she wrote is a test plan for you, then you might have a serious problem. What are your rights to use this test plan in the future? To make copies of it to give to your staff? To create other test plans following the structure and style of this test plan?

Your contract with the outsourcer should explicitly and carefully address the ownership of work products. If it doesn't, and if you are lucky, then a court might rule that the outsourcer's work was "work for hire" and copyright transferred to you anyway. But if the outsourcer is not your company's full-time employee (which the outsourcer will almost certainly not be), then the work for hire doctrine might or might not apply to your situation. When you are drafting the contract with the outsourcer, it is unwise to rely on this doctrine. Instead, be explicit.

Under what circumstances should the outsourcer own materials that he created while providing services to you? Many of them would strenuously argue that they should own any generic materials (things that can be reused across several projects, several companies). They need this to build their efficiency over time. But you should ask for a royalty free, fully paid up license to use these materials in any future project.

Examples of the types of work products that are at issue here include:

- Computer programs
- Test automation scripts
- Test plans
- Tables and matrices for describing collections of test cases
- Course slides that the outsourcer developed and used to train your staff or the outsourcer's staff during your project (at your expense)

If this material is prepared specifically for your project, and is suited only to your product, there's rarely going to be a problem. But to the extent that the outsourcer can use the same material for other clients (after scrubbing anything that might reflect your trade secrets), the outsourcer will or should want to keep rights to it.

The feelings on both sides of this issue run hot. On one side, you have companies who spend a fortune to obtain a series of database queries and reports, a special effect for a movie, a framework program to run their test cases, etc. They feel cheated when the outsourcer or consultant reuses this material on the next assignment. After all, the next guy gets the benefit of all of this expensive research and development, and pays almost nothing for it.

On the other side, the outsourcer or consultant builds a library of materials over time, and uses them whenever they are appropriate. This keeps client costs down. And so many projects are very similar (e.g. test harnesses can be reused in a wide range of circumstances). If the outsourcer or consultant agrees to forfeit its ability to create work products that are very similar to previous ones, it might soon go out of business.

I feel very strongly that the consultant and outsourcer should develop a generic library of tools and solutions, and should use them on project after project. When I interview outsourcers, I ask to look at the types of reusable materials that they have available. As a customer, I save money when the outsourcer uses these tools. The tradeoff that I recognize is that some of the custom work that I pay for will go into the toolkit for the next customer.

Not everyone shares my feelings, and this issue is a common dealbreaker.

When you run into an impasse, you might be able to break out of it by working at a finer level of detail. Perhaps some items can be generic while others cannot be. What distinguishes them? The amount of money you had to spend getting the item developed? The extent of involvement of your staff in the analysis, development, or review of the item? The extent to which the technology at hand is central to your business, not just something that you find useful?

In my experience, the engineering staff on both sides are more likely to find common ground than the lawyers. The lawyers tend to get locked into positions, and often don't know enough about engineering to know how to safely break out of those positions while still serving their clients.

Additional Standard Contractual Issues

This section considers other significant issues that are normally resolved during contract negotiation. The difference between these and the ones mentioned above is that lawyers are likely to see these as part of

the deal, while testers / engineers often are less interested in them. The section includes discussion of nondisclosure (confidentiality) and noncompetition clauses, conflicts of interest, your ability to hire staff from the outsourcer, liability (including indemnification) for bad work and for breaches of other types of warranties, and the process of terminating the contract.

Confidentiality and publicity

Obviously, you don't want your outsourcer leaking your trade secrets. Nor do you want an outsourcer to feel that it can claim to the press that it has a very close relationship with you just by quoting the work it has done for you. Those claims can generate bad publicity for you and might generate liability.[3]

On the other hand, the outsourcer needs the ability to market its services, to quote examples of its work, to publish case studies. Variations of the following clause are sometimes proposed and seem reasonable to me:

> *Company recognizes that Consultant maintains her professional knowledge and reputation by attending professional conferences and publishing materials that help develop the state of the practice of software quality improvement. To that end, Consultant may publish documents of general applicability that do not disclose Confidential Information of Company. Such documents may describe the services performed for Company by Consultant and results obtained, so long as the identity of Company and of individuals associated with the project are suitably disguised. Documents that Consultant develops at the Company or using any of the Company's facilities are subject to the Company's approval. Such documents will be the property of the Company, but Consultant will have an irrevocable, nonexclusive license to publish them and to create and publish derivative works based on them and to license others to publish them.*

Warranty for services

If you develop a product, you may be asked to provide a warranty that the product is reasonably reliable. Standard-form contracts for custom-programmed software often contain warranty clauses.[4] Here's some language that blends a few contracts that I've seen:

> *Contractor warrants that the software will conform to the specifications incorporated in this Agreement. Contractor will, without additional charge to Client, make such modifications to the software as may be necessary to correct any defects reported to Contractor by Client for a period of one (1) year after the acceptance date. If Contractor is unwilling or unable to make the required modifications then Contractor will reimburse Client's reasonable expenditures for obtaining the required modifications from some other Contractor of Client's choice.*

---

[3] For example, a test lab announces to the public that they have tested your product and it is absolutely compatible with a certain specification. Based on this claim, a customer buys the product and finds out the hard way that there are remaining incompatibilities. The customer demands a refund and damages from you, saying that "you" (the test lab that worked with you) made public claims that were false. At a minimum, you have a very unhappy customer who feels that you lied to him, or that your business partner lied to him, even though the lab is not your business partner, is not authorized to speak for you and you would never have made the claim that it made.

[4] For example, look at the warranties in Raysman and Brown (1984, supplemented 1996), "8.22 Form: Software and Equipment Development Agreement." See also W. Hancock (Ed.) *Corporate Counsel's Guide to Software Transactions*, Business Laws, Inc., 1995, Form 223, "Programmer Agreement for Individual Consultants."

This may be appropriate for a programming agreement, but if your company uses a battery of standard clauses in its software-related contracts, this clause might also creep into your standard contract for testing services. This doesn't make sense. Your outsourcer can't control the quality of the software. They find the bugs, but it's up to you to fix them (or not). And will you really allow the outsourcer to do as much testing as it considers necessary for it to warrant the quality of a product, no matter what the impact on your cost or schedule? Probably not.

Outsourcers are more likely to make guarantees that focus on what they do, rather than on the quality of the released product. (Of course, a lab that sellers certification services is leaving itself open to complaints that product does not live up to the certification.)

Ocampo, Curtis, & Moss (1996, section 4.01) suggest the following language and provide some methods for resolving disputes about what is meant by "generally accepted industry standards."

> *Contractor warrants that the services will be performed consistent with generally accepted industry standards.*

The outsourcer can only give this warranty if you will give it time and budget to achieve that level of performance. And the outsourcer will face malpractice risks as the IEEE persuades more states to license software engineers.

If you pay the outsourcer by the hour, maybe a good clause looks like this:

> *Tester will provide effective, efficient testing services and will strive diligently to thoroughly test the program. Client and Tester understand and agree that Client faces tradeoffs between its needs for high product quality, timely project completion, and limited development cost. Tester and Client will work together to determine the extent and depth of testing that can reasonably be achieved in light of Client's other constraints and the Software's design and reliability.*

Indemnification

Indemnification clauses come up most often in contracts between large customers and smaller service providers. The large customers insist on them, demand them, say that they would never ever sign a contract that didn't have them. These same customers are often software publishers or developers themselves, who would never dream of agreeing to such a clause in a contract in which they were the vendor. Pointing out this inconsistency is a useful negotiating tactic when the indemnification clause begins to look like a dealbreaker.[5]

An indemnity agreement involves a promise that, if a specified event happens, the indemnifier gives the company money. The most common form of indemnity agreement is called "insurance." Consulting firms should think carefully about whether it makes any sense whatsoever to agree to pay a client a nickel for anything that is not their direct, personal fault or that happens under circumstances that are beyond their control. As a consultant (or an attorney representing consultants), when I see one in a contract, I usually cross it out completely. If the company won't accept that, I reword it to something that gives the company narrow protection against carefully specified risks.

Here's an example of a troublesome indemnity clause:

> *Contractor agrees to indemnify and hold harmless Company and its directors, officers, and employees against all losses, liabilities, judgments, awards and costs (including legal*

---

[5]. Ocampo *et. al*, *Negotiating and Drafting Software Consulting Agreements*, Glasser LegalWorks, 1996, recommend this approach in their section 6.03.

fees and expenses) arising out of or relating to (1) services performed by Contractor; (2) breach of this agreement by Contractor; or (3) the use of subject matter by Contractor or by Company, to the extent that such subject matter is provided by Contractor, in which Contractor knows or reasonably should know, that others have rights.

If the outsourcer tests a program, any failures of that program relate to the services that they performed. So, if your program has a bad bug and your company gets sued, the outsourcer will have to pay. But what if the outsourcer had actually found and reported this bug to you, but you shipped the software anyway, and were then sued? Should you be able to go to the outsourcer for indemnification? The plain language of the contract says yes, but surely this would be unfair.

What about hard-to-find bugs or bugs that came into the program late in development? What about bugs that the outsourcer missed because the schedule was too tight to run all the tests the program needed? Should the outsourcer be liable for these?

Next, suppose that the outsourcer actually is at fault. Suppose that she missed a bad bug, that a different test group might have found. This clause says the outsourcer pays everything—all losses, liabilities, judgments, attorney's fees, court costs, etc. Maybe -- *maybe* -- she should owe the company *something* in this situation. But *everything*? That's ridiculous. The outsourcer didn't make the bug. She didn't set the development schedule and she probably didn't set the testing schedule. She didn't set the expectations of the now-angry customers by deciding how to advertise the program, what to say on the box, what to say during the sale, or who to sell it to. She didn't fail to take care of the customers when they called the company's technical support center. Even if she is at fault, so is the company, and probably much more at fault, in more ways, than her. Why should the outsourcer pay everything and the company pay nothing?

Here's another common variation on the all-encompassing indemnity clause:

Contractor shall defend, indemnify, and hold harmless Consultant from any losses, liabilities, damages, demands, suits, causes of action, judgments, costs or expenses (including court costs and attorney fees) resulting from or directly or indirectly arising out of or in connection with this Agreement and the transactions contemplated hereby, including but not limited to breach of any representation or warranty made herein.

Company shall have the right to approve any counsel retained to defend any demand, suit, or cause of action in which Company is a defendant, such approval not to be unreasonably withheld. Contractor agrees that Company shall have the right to control and participate in the defense of any such demand, suit or cause of action concerning matters that relate to Company, and that such suit will not be settled without Company's consent, such consent not to be unreasonably withheld. If, in Company's judgment, a conflict exists in the interests of Company and Contractor in such demand, suit, or cause of action, Company may retain its own counsel whose fees shall be paid by Contractor.

This is classic insurance. The outsourcer pays even if there is no bug and no fault. If someone sues, whether they win or lose, the outsourcer will hire the lawyer, defend the company in the lawsuit, pay for the company's lawyer so the company can help the new outsourcer to defend the lawsuit, and you pay for everything else. Their responsibility is triggered by anything "in connection with this Agreement."

Actually, this isn't *quite* classic insurance, because the standard General Commercial Liability policy doesn't insure a company against warranty claims for defects in its own products. Under this clause, the outsourcer does. Furthermore, insurance policies have limits -- the insurance company pays up to the limit and not beyond. Under this clause, the outsourcer pays everything.

Note also that this second example specifies a "breach of any representation or warranty" as a specific basis for holding the outsourcer accountable. The contract might involve a wide range of warranties, making this indemnification broadly dangerous.

Here's another example of troublesome contract language:

> *Contractor shall indemnify Company from all claims, losses, and damages which may arise from the breach of any of Consultant's obligations under this Agreement.*

What are the "obligations"? The contract's warranties certainly define some of them. Maybe there are others too. The indemnification clause serves as a blank check—unlimited liability—and the range of events that could trigger the writing of the check is vague. The wise outsourcer will be very reluctant to agree to this clause.

Most service providers refuse to indemnify clients for claims involving defects in the tested products. Some will provide defect-related indemnification, but will demand some basic protections:

(a) *Proportionate liability*: They only pay in proportion to the degree to which they are at fault.

(b) *Prompt notification*: You must notify them promptly—within days—after you are told of a claim that they might be made accountable for.

(c) *Participation*: They have the right to participate in the defense (i.e. they should be allowed to send their lawyer to all settlement meetings, mediation, arbitration, trial, etc.).

(d) *Approval*: They want the right to approve or disapprove any settlement agreement. This is reasonable because they must pay all or some of the amount of any settlement. Perhaps they will agree to not unreasonably withhold their approval of a settlement or to continue the defense at their own expense if they reject a reasonable settlement.

(e) *No Loss, No Liability*. They refuse to pay anything if the company wins the case. Their indemnification involves only their defective work They pay only for a settlement that they agreed to or a court judgment or arbitration decision that says their company is at fault (and only to the extent of their proportional liability as determined by that court or by negotiation with you.).

Consultants sometimes find it easier to get agreement on notification, participation, approval, and loss-only liability than to get agreement on proportionate liability. But they have a strong argument that anything else is fundamentally unfair. The notion that defendants should only have to pay damages in lawsuits up to the degree that they are at fault is one of the most popular and most sensible "tort reforms" pushed by the American business community.

Companies ask outsourcers to provide other types of indemnification, against other types of risks. For example, they might want the outsourcing company to be fully insured in case one of its staff causes a personal injury to an employee of the client company. They might want assurances, backed by an indemnification, that the outsourcer is managing the payroll tax deductions of its staff in ways that will not come back to haunt the client company. I discussed these in Kaner (1997a). These can be entirely reasonable, and if they are broken out separately in the contract, they might be very reasonably worded.

Indemnification for Intellectual Property Infringement

This last variant is the most reasonable one for a company to demand and the one that the outsourcer might find hardest to escape. Ocampo *et. al* (1996, Chapter 5, "Infringement Liability.") provide a long, thorough version of this clause and an extended discussion. A common short version reads:

Contractor will indemnify Company against liability to third parties resulting from claims that the software developed pursuant to this Agreement infringes on or violates any patents, copyrights, or trade secrets of such third parties.

According to Ocampo *et. al*., indemnification for patent infringement is often a negotiation issue because you might not realize that a technique that you're using has been patented. Also, in American contracts, trademark and trade secret infringement indemnification clauses are usually restricted to the trademarks and trade secrets that are held in the United States.

I will sign these clauses, and I advise clients to agree to these clauses, so long as they provide for the basic safeguards (prompt notification, participation, approval, no-loss-no-liability). I've deliberately dropped "proportionate liability" from this list because it is rarely relevant here. The client company is rarely at fault if the consultant uses trade secrets or copyrighted material that he got somewhere else. However, the outsourcer should not indemnify the company for its use (or the outsourcer's use, as part of its work for the company) of material that was supplied to the company by someone else.

Here's language that I sometimes use:

Contractor agrees to cooperate in the defense of Company in any claim made by a third party that the work performed or work products delivered pursuant to this Agreement infringe on or violate any patents, copyrights, or trade secrets of such third party. Contractor agrees to indemnify Company against liability to third parties from any settlement or final judgment award, including without limitation reasonable attorney's fees and other expenses awarded, that arise from the use of subject matter by Contractor or by Company, to the extent that such subject matter is provided by Contractor, in which Contractor knows or reasonably should know, that others have rights. This indemnification is contingent on Company providing prompt written notice of such a claim to Contractor, and granting Contractor the right to participate in the defense of any such claim, and the right and opportunity to approve or reject any settlement of any claim for which Company will seek indemnification from Contractor.

Managing The Ongoing Relationship

This at the day-to-day relationship between your company and outsourcer. You must supervise the outsourcer's work, evaluate the bug reports you get back, check whether the tools are useful, the schedule is being met, the usual project management issues.

In developing this section, I gave particular attention to two issues. First, I think it is essential that you appoint a liaison with the outside test lab, someone who will actively work with the outsourcer and resolve the day-to-day issues. Second, I think that there are many circumstances under which it is more valuable to bring the outsourcer's staff onsite than to farm the project out to the outsourcer's lab. This has been my strong preference for years. The Software Test Managers Round Table worked on this issue at length in their meeting of October 31 and November 1, 1999, and opened my eyes to a wider range of factors to be considered in this decision.

I'm not satisfied with the treatment of performance standards in this outline. In general, in outsourcing relationships, you can establish and agree on Service Levels, and evaluate performance according to those agreements. Service level agreements typically involve unambiguous quantitative reports.

Unfortunately, I think that we have trouble measuring the quality of software or of software testing. Difficult side effects show up when we try to measure (and manage on the basis of these measurements of) such things as extent of testing (Kaner, 1999). My caution about objective-looking "measures" is reflected in the outline. We need to develop better measures but for now, I think that the evaluation of the ongoing work of the outsource test lab will have to be largely qualitative.

Testability Issues

How do you enable the outside test lab, to give it a chance to be more efficient and more effective? This section looks at several thinks that the in-house test group or programming group can do or deliver to the outsourcer.

Other Standard Contracting Issues

Not much here of interest to engineers, but you just can't have a contracting checklist without raising issues like severability, choice of law, choice of forum, assignment, and integration of terms. It's just not done.

Enjoy the checklist!

--- Cem

Outline for Software Testing Outsourcing

Cem Kaner, J.D., Ph.D.

Note on the Formatting of the Outline

Deep outlines need special formatting to help the reader not get lost. This is a 7-level outline. I distinguish among the levels as follows:

H. LEVEL 1

1. Level 2
   a. Level 3
      A. Level 4
         1. Level 5
            a. Level 6
               A. Level 7

A. DECIDING THAT YOU WANT TO OUTSOURCE

1. **Potential benefits of outsourcing.**
   a. **Lower, control, or limit testing costs.**
   b. **Expand and contract your testing capacity as your group's workload grows and shrinks.**
   c. **Use the outsourcer to train your staff.**
   d. **See how an expert uses a new tool or method.**
   e. **Certify your software.**
   f. **Delegate (outsource) unpopular tasks.**
   g. **Clarity: You and the outsourcer can agree on what you're going to get for your money.**
   h. **The outsourcer might have better testing project management experience than you do.**
   i. **The outsourcer might be able to start sooner than you can.**
   j. **Helps you make visible many of the currently hidden costs of testing.**
2. **Risks of outsourcing.**
   a. **The outsourcer may drive you to change your processes in bureaucratic ways that don't help the company.**
   b. **The vendor may disrupt all of your planning by drastically underestimating the work it will have to do for the project and then blaming you for every overrun. (This is a real problem, the vendor underbids to get the job, and then needs an excuse to significantly increase what it will actually receive.)**
   c. **The outsourcer's processes may be sloppy and the testing may be much more shallow than the work you would do in-house.**
   d. **There may be huge cost overruns.**
   e. **The work may be done poorly.**
   f. **The outsourcer will not understand your market as well as you do and will miss or underrate important issues.**
   g. **Security leaks.**

h. **You may pay for services without contracting to receive title or license to the intellectual property (such as test plans or test automation suites), making you addicted to this outsourcer or unable to reuse the materials that were created.**
i. **Key staff of the vendor's might be overcommitted, switching attention between several projects (only some of which are yours).**
j. **If there is an interruption in your project, the staff who were assigned to your project may move on, costing you a huge and unexpected learning curve expense.**
k. **The outsourcer may only tell you what you want to hear.**
l. **The outsourcer may not work as hard as your staff would have.**
m. **Communications problems may waste time and cause you grief:**
    A. *The vendor may alienate your programmers.*
    B. *The vendor may alienate your staff.*
    C. *The vendor's reports may be so poorly researched that your staff spend inordinate time trying to replicate and extend them.*
n. **The vendor may disparage your staff or try to prove they are incompetent as a tactic for gaining more business from your company. (Your company may be convinced to lay off lots of testers, in order to use the outsourcer.)**
o. **Your management may treat outsourcing as an excuse to create less documentation or work more sloppily (after all, no in-house tester is going to complain that these problems are slowing them down when the testing has been sourced out.)**
p. **You may become a captive of the outsourcer. On a product with a steep learning curve, the outsourcer's staff may become the only people who know how to test your product.**
q. **There are human resources risks and legal risks associated with reducing your staff and replacing them with an outsourcer. See Hancock's *Corporate Counsel's Guide to Outsourcing: "Employee Relations Aspects of Outsourcing."***

3. **Benefits to the outsourcer.**
    a. **Money: your payment for the outsourcer's services.**
    b. **Opportunity to learn a new technology.**
    c. **Opportunity to develop an ongoing relationship with your company.**
    d. **Ability to advertise an association with your company.**
4. **Circumstances that motivate outsourcing.**
    a. **Your management just told you to outsource the testing.**
    b. **You're running too many projects and need to focus.[6]**
    c. **You don't have enough staff and can't recruit.**
    d. **You have a headcount limit but need to add staff to a project *somehow.***
    e. **You can't start the job fast enough. (Outsourcers are used to starting on a dime.)**
    f. **You can't do the job fast enough.**
    g. **Your company doesn't have a test group. You are a project manager who has just been told to arrange for the testing.**
    h. **You're missing needed expertise.**
    i. **You need better documentation from the programmers / project managers but can't get it for your team. Everyone understands the communication problems of remote resources, so you think of using an outsource staff as an excuse for demanding better documentation.**
    j. **You need an independent advisor to resolve disagreements.**

[6] This can be an excellent reason to outsource, but don't forget that you still need to dedicate a staff member, part-time or full-time, as a supervising liaison.

k. **You have a contractual or legal obligation to obtain independent testing / certification.**
l. **You're missing needed equipment or software (such as hardware needed for configuration testing, or testing tools).**
m. **You want to farm out maintenance-oriented testing for an old-technology product that no one on your staff wants to work on any more.**[7]
n. **Someone in management wants the independent judgment of an outsourcer because they believe that the outsourcer will be "more reasonable" (more willing to roll over on quality issues) than in-house testing staff.**[8]
o. **Some yahoo in senior management "knows" that outsourcers do it faster, cheaper, better. ("Let's quit worrying about testing.")**[9]
p. **"Avoid confronting the effects of bad management decisions regarding in-house information processing." (Halvey §16.02)**
q. **You want to try outsourcing as a pilot project.**

5. **Favorable conditions for outsourcing.**
   a. **The project is well organized and easy to outsource.**
   b. **You're testing Release X.01 and you have a good test plan from X.00.**
   c. **You're testing an OEM release (custom release with minor changes to support a new operating system, printer, video card, or computer) after the base version is tested and shipping.**
   d. **You can delegate part of the project, and that part is fairly standard or well understood by the outsourcer. (Such as load testing or smoke testing.)**

6. **Think about the services to be provided to you**
   a. **Identify what is good and bad about your current system. What do you want to keep and what should you delegate?**
   b. **Scope of the testing**
   c. **Type of testing**
   d. **Supplement your in-house testing**
   e. **Other products or services than testing**

B. FORMING THE RELATIONSHIP

1. **Agree on your schedule**
   a. **You want to ship it** *next Tuesday***?**
      A. *Genuine rush jobs cost lots of money and require an outsourcer who can ramp up immediately. Do you* **really** *need it next Tuesday?*
      B. *What contingencies if you don't finish by Tuesday. Will the outsourcer's staff and labs still be available to you? Will they work overtime? For how long?*
      C. *If speed is essential to you, what circumstances allow this outsourcer to test more quickly than you can? You need facts, not general reassurances.*
   b. **Are you being told to use an outsourcer to make it "possible" to achieve a fantasy schedule that the rest of your company can't achieve? This distorts your cost structure:**
      A. *Rush charges to ramp up.*

---

[7] This is a tempting reason to outsource. But how will you ensure that the outsourcer does an adequate job of testing? If its work is inadequate, will you be able to get enough attention of management to deal with it?

[8] If this turns out to be the driving motive for selecting an outsourcer, and if the outsourcer plays along, you may want to file a report to senior management at some point that documents the negative value provided by the outsourcer. (1) Plan to document the outsourcer's stupidest recommendations. (2) Plan to retest all bugs filed by the outsourcer (look for inadequate analysis and inadequate follow-up). (3) Plan to conduct some sampling tests to see if areas are being skipped by the outsourcer.

[9] Plan to conduct some research on defect rates in the field. Did the outsourcer did a good job of finding defects (as measured by what undiscovered bugs are still present) or not?

B. Additional charges for going beyond the planned number of cycles of testing.
C. Additional charges / penalties for tying up key staff of the outsourcer past the agreed end date for the project.
D. Additional ramp-up costs to bring in new staff. This problem can arise if key staff of the outsourcer were committed to another project. They were temporarily assigned to your for a short term, because you promised that it would be a short project, but now that you're running overtime, you can't keep these people.
E. All the inefficiencies of a no-time-to-plan-just-do-it style of testing.

c. If the schedule is unreal, maybe you should use the outsourcer as a neutral expert who can scope out the project and explain the schedule difficulties to your management.

d. Planning For A Successful Relationship

2. Work through the costs

a. Develop contract terms that manage costs.

A. Basis for compensation

1. The compensation issue is the most important issue in the contract. If you accept a fixed-price bid, the contract must be clearer about the scope of work paid for (because the outsourcer's incentive is to prove that everything is outside of the scope) and about the deliverables owed each way. The contract resolution procedure **must** also be clean because there will be conflicts about scope, pace, and deliverables.
2. Pay by the hour
    a. You control the quality of the job. If you need more work done, different work done, etc., then you pay for it and you get it.
    b. The outsourcer has an incentive to stretch out the job.
3. Flat fee
    a. You have a cap on the costs.
    b. The outsourcer has an incentive to take shortcuts. You should be more interested in getting warranties from the outsourcer, as a counter-balance.
    c. If the project definition changes, you will have to pay extra. With extra charges for "scope changes" and other changes, a "flat fee" can cost more than an hourly job for the same amount of work.
4. Flat fee with incentives and disincentives
    a. For example, you pay extra if the testing is done on time or early; you pay less for projects that get finished late.
5. Piece-work
    a. Pay by the bug? (Not a good idea.)
    b. Pay according to the number of function points or KLOC to be tested? (This is essentially the same as a flat fee situation.)
    c. Pay per configuration tested (e.g. $X for each printer tested). This is more sensible. It only covers this one aspect of the project. In your negotiations, plan for some retesting. Also, if you want compatibility reports for tech support, negotiate for them and expect to pay for them.
6. Pay by the milestone or on completion of tasks
    a. Your goal is to pay only for services that you actually receive.
    b. You want to pay for actual services rendered, not for tasks that are only more-or-less complete.
        A. Mylott (p. 150) "If you pay for poor service then you subsidize it. Your vendor will undoubtedly argue that you received some value even from inadequate service. So what? Why pay for poor service even if you did receive some value from it? . . . Don't be surprised if your vendor rejects this type of approach." If it's not what you contracted for, and not what you would contract for today if you were doing the deal, then the outsourcer has provided you with negative value. You wanted to pay for better service and you got stuck with this.

c. But what if the delaying factor is the lateness of your software or the slowness of your review?

B. Payment for incidentals

1. Printing of special reports, binding, shipping, telephone, etc.
2. Research costs.
3. Travel expenses
4. In general, what is your extent of control and liability for the outsourcer's out of pocket expenses?
5. What about the outsourcer's office and overhead expenses? Some law firms bill for these types of expenses and other consulting businesses have been picking up the practice.
   a. If they are raising their effective fee by passing through more costs, are they going to lower their hourly fee?
   b. Is it easier or harder for you to manage these expenses within your company's internal budgeting system? For example, do incidentals show up on the same department's budget that the outsourcer's fees show up? Do these costs count against your budget or against your company's general overhead?

C. Taxes

1. Who pays the taxes if taxes are levied on the software licenses or on the transfer of licenses or on the services provided by the outsourcer?

D. Invoicing

1. What triggers an invoice
   a. Periodic bill? (Monthly? Weekly?)
   b. Milestone achieved? (Who signs off that it has been achieved?)
   c. Out-of-pocket money spent (invoice for reimbursement, as expenses are incurred)
2. When is payment due?
   a. 10 days after invoice? 30 days? 45 days?
3. What level of detail is required with the invoice?
   a. Hourly timesheets from outsourcer's staff?
   b. Task-based timesheets showing how many hours spent for what tasks?
   c. Summary record listing people and hours?
4. Even less detail?

E. Auditing

1. Do you have the right to audit the outsourcer's records to determine the accuracy of invoices submitted?

F. Most favored nation clause

1. Assurance that you are being charged no more than any of the outsourcer's other customers.

G. Changes in fees

1. When can the outsourcer's rates be renegotiated? (In a long-term contract, this is inevitable.)
2. Is there a cap on the increase?

b. Develop (with the outsourcer) a project plan that shows times and costs.

A. Does the outsourcer agree to be bound by the plan?

B. What circumstances would cause change in the plan in ways that affect your costs?

C. What control do you have over those circumstances?

c. Contract to ensure that the right resources are applied to your project.

A. See the discussion of qualifying the outsourcer.

d. What are your quality-related expenditures?

A. Which of these will be outsourced?

1. What do they cost you now?
2. How much will you save on them?

B. Which will not be outsourced?

1. What do they cost you now?
2. How will outsourcing other tasks affect the costs of what is left? Will you lose or gain efficiencies?

C. What additional costs are you taking on?

1. Supervision of an outside party.
2. Communication costs.
3. To achieve the same quality of testing as you do in-house, will you have to provide the outsourcer with better product documentation?
   a. Who on your staff will develop the documentation?
   b. Will you be able to get needed specifications from the programming staff?
4. Cost of changing product development cycle to accommodate outsourcing.
5. How will you estimate and factor in the probable level of cost associated with relationship difficulties, such as copyright infringement, trade secret leaks, dispute resolution, and so on.

D. Are you achieving savings simply by doing less testing, especially by dropping certain types of tests (such as customer scenario tests)?

e. No outsourcer will work for you at a loss for very long, so how are the outsourcer's economies achieved?

A. Always question costs that seem too low.

B. If the outsourcer's staff are cheaper than yours, will they be as good?

C. If the outsourcer agrees to a fixed-price bid, it will find a way to add charges when the programming falls behind schedule or the feature set changes.

D. Is the outsourcer playing a game with you where you sign a long-term contract today at a low initial price but with higher costs later? The worst part of this is the blame game, every little change creates a re-estimation and all those changes are all your (or your company's) fault. You get caught in the middle between the outsourcer on the one side and the programming team and project manager on the other.

E. Watch out for hidden machine costs. Some outsource agreements limit the number of machines used for testing. Find out if extra machines mean extra money.

f. You can lower costs if the outsourcer will be more efficient than you are or has appropriate equipment on hand. Have the outsourcer explain, specifically, in what ways it is unusually efficient. How will those efficiencies save time on your project?

A. You can lower hardware testing costs

1. If the outsourcer has the right equipment and has efficient lab procedures, and will assign the right people and resources to your project.
2. If the outsourcer is willing to buy equipment for compatibility testing that will soon become outdated, and to spread this expense across many projects.

B. You can lower test creation costs:

1. If this outsourcer has a library of standardized test materials that apply to your application.
2. If this outsourcer has relevant test tools (such as tools for load testing) and the expertise to use them.
3. If the outsourcer has efficient processes for creating test cases.

C. You can save test research and development time if the outsourcer has a deep understanding of the type of environment that you are testing in.

D. (Some) outsourcers have standardized automated testing frameworks, to generate test suites quickly.

E. (Some) outsourcers have well-trained staff who find bugs quickly.

3. Define the services

a. The scope of the testing

A. *Entire testing department ("We give up on testing.")*[10]
B. *Entire platform ("We give up testing Macs.")*
C. *Entire product line ("We give up testing word processors.")*
D. *Entire product (Buggyword)*
E. *Product release (Buggyword 5.0)*
F. *Maintenance release (Buggyword 5.01)*
G. *Entire language (All of our French-language products.)*
H. *Additional testing to supplement the effort of your group. (See "What type of testing do you want done?")*
I. *Required independent test. (Whatever extent of independent testing is required by contract or regulation.)*

b. Type of testing

A. *Audit your work.*
B. *Beta testing*
C. *Device (printer, video, etc.) compatibility testing.*
D. *Documentation (accuracy of the user-level docs)*
E. *Feature benchmarking.*
F. *Function testing. (Basic black box testing, probably to a spec.)*
G. *Installation testing*
H. *Load testing*
I. *Localization or international testing.*
J. *Performance benchmarking.*
K. *Quality assessment or measurement.*
L. *Risk analysis*
M. *Sanity check (look for obvious holes in your work).*
N. *Security testing.*
O. *Software interoperability testing.*
P. *Stress testing*
Q. *Usability testing.*
R. *User scenario testing, i.e. create complex test cases from real-life uses of similar products.*
S. *Validate against standards or government regulations.*
T. *Web (or other platform) version of a popular PC product?*
U. *Web security. Be very careful if you choose this one.*

c. Other products or services than testing.

A. *Train your staff. What do you want to learn from this?*
   1. Testing techniques.
   2. Automation techniques.
   3. Test planning techniques.
   4. Show us how to create a test plan.
   5. Show us how to develop test suites.
   6. Risk analysis.
   7. Standards.
   8. How to use a specific testing tool.
   9. How to test a product based in a specific technology (e.g. client/server, or web-based, etc.).
   10. Obtain general-purpose test planning documents.
   11. Reports on device compatibility that your tech support staff can use.
   12. Reports on bugs/workarounds that your tech support staff can use.
   13. Approaches to metrics.

[10] This might be a recipe for disaster.

B. Develop automation.
1. Write a test harness.
2. Write the test code.

C. Documentation
1. Develop a test plan.
2. Develop a process description.

D. Consulting
1. Help us plan our automation strategy.
2. Help us evaluate our processes.
3. Help us plan a measurement approach.
4. Help us develop or choose a bug tracking system.

E. Certification
1. ISO 9000 auditing.
2. Compatibility with some other standard.
3. Certified compatible with some third party device.
4. Certified as passing some third party test series.

F. Other non-testing services.
1. Build management.
2. Bug management (act as a bug tracking service bureau).
3. Evaluate deferred bugs (pre-shipment; independent appraisal of their importance.)
4. Compute and report metrics.
5. Recruiting.
6. Trade show support? (Test groups often provide this type of support because they know the program, and how to work around its problems. Do you need this from outsource staff? Who else will do your demos at the trade shows?)
7. Is the QA staff also the IT staff for testing? While they are setting up their own servers and set up their own machines, they are not testing. This slows down test cycles. Find out if the functions can be separated and at what cost. Maybe you can outsource system administration and keep testing?

d. What services are excluded from the agreement?

4. Thoughts on other benefits that you want to realize by outsourcing

a. Evaluate your goal of expanding and contracting your testing capacity.

A. Is it realistic to think you can outsource certain tasks?
1. What is the learning curve for developing your type of product?
2. What key knowledge do you lose in-house if you delegate the task?
3. Do the tasks involve critical business secrets or know-how?
4. Are these tasks intimately connected with other projects that you will keep in-house?
5. Do you have the management depth and time to manage offsite development?

B. Qualify the outsourcer (make sure they have resources and can assign them). (See below).

C. Manage the costs of training the outsourcer's staff (after someone learns your systems, will they be assigned to subsequent projects of yours?)

b. Using the outsourcer to train your staff.

A. Training by example.
1. Have your staff work side by side with the outsourcer's staff so that they can learn in detail.
2. To create the equivalent of apprenticeships to the outsourcer's senior staff, you must budget their time and your staff's time to allow for the informal training.

B. Training by instruction.

1. Arrange for frequent seminars for your staff, for the outsourcer to provide training on how / why they're doing the testing the way they're doing it.
2. Build training requirements into the contract, so the outsourcer knows this is a priority.

C. How will you assure that you get the knowledge transfer that you're paying for?

c. Seeing how an expert uses a new tool or method.

A. Will the outsourcer's expert be assigned to your project?

B. Are you arranging for training time, to transfer skills with this tool to your staff?

d. Using an independent advisor to resolve disagreements or identify issues.

A. Be clear in the contract that this is your requirement or you will get testing, not advice.

B. Is this independent advisor competent to give you this advice? What does she know about your market? Will she recognize a productivity-killing usability error?

e. Certification

A. Is this outsourcer recognized (certified, licensed, registered) as competent for this task?

B. Does your understanding of the outsourcer's duties match the outsourcer's?

C. Certifiers' approaches vary. Interview this person or group carefully to ensure that their interpretation of the relevant standards and auditing processes are compatible with your objectives.

D. Specify the scope carefully. Is this a certification of the product or the process? Is it based on late-stage testing of a finished product, or on a broader review of the development processes?

E. How will you manage the risk that this organization will declare that your product is not good enough?

f. Delegating unpopular tasks.

A. What are the career implications for the in-house staff who co-work with the outsourcer on this project?

B. If this project is unpopular because it is on the road to failure, who will the in-house scapegoat be?

C. Are there any other in-house human resources issues associated with this outsourcing?

D. Will the outsourcer do the undesirable work in the way that you want them to? If not, is this a problem?

E. Is this an excuse to lay off staff?

g. Trying outsourcing (pilot project).

A. Make sure that you capture all of the relevant in-house and outside costs.

B. Do some late auditing of the quality of the outsourcer's work. Get data on the actual costs and benefits of the effort.

5. Qualifying the outsourcer

a. Visit the outsourcer's facility (if their staff will work there)

A. Look over the facility.

B. Meet the people (the working testers, don't be distracted by the executives).

C. Look over their standard test materials.

D. Look at their equipment.

E. Look at their security.

1. Get a tour of the lab.
2. Whatever you can see, others can see.

b. Visit another outsourcer's facility

A. Until you are experienced in dealing with outsourcers, use interviews to get to understand the variation across companies.

c. Will the outsourcer assign the right staff to your project?

A. Does the outsourcer have available staff who have the skills you need?

1. How do you know?
2. Have they published any books or white papers that you can read?
3. Will they show you the generic test materials / tools that their staff have created?
4. How heavily booked is the lab? How many people (and which ones) are actually free or will be free within a few days?
5. If the outsourcer has the right people on staff, will they be assigned to you?
6. *Beware of bait and switch.* If you have interviewed specific staff of the outsourcer, will these be the people who work on your project?
7. If the outsourcer has access to a pool of skilled talent, will they be hired for your project?
8. If you need specific expertise, will the outsourcers find and assign the right types of experts to your project?
9. Do you have control over which of the outsourcer's staff do or do not work on your project? Can you require the outsourcer to remove / replace one of its staff from your project?
10. What kind of training program does the outsourcer have? When they bring on new people, how do they get them up to speed quickly?
11. How well do the vendor's staff speak and read English?
12. Can their testers be bonded? Do they need to be Series 6 or 7 for financial applications?

B. Professional performance clauses. Does it make sense to require that:

1. The outsourcer's staff will perform all services in a professional manner and that all services will be performed by qualified and properly trained personnel.
2. The outsourcer's staff will have computer science (or equivalent) degrees from accredited universities.
3. The outsourcer's staff's services will meet the highest professional standards. (Watch out for this one. Are you obligating the outsourcer to meet standards, such as various IEEE standards, that you might not be interested in following nor willing to pay for?)
4. In some states, beware of resistance based on malpractice liability risks. For example, Texas now licenses software engineers. This creates risks.
    a. It is unclear how courts will interpret license-related standards and therefore unclear what the business risks associated with licensing will be.
    b. It is unclear what standards are to be followed in order to achieve "competence" performance. It is unclear how long it will take to establish those standards.
    c. I think that any sane outsourcer who offers services to a Texas corporation will insist on a contract clause that specifies that the services provided are not software engineering and that the services offered are not to be evaluated against professional standards for software engineering.

d. Will the outsourcer assign the right equipment and tools to your project?

A. Does the outsourcer have the right equipment and tools?

1. How do you know?
2. If the outsourcer has to obtain new tools or equipment, will they be obtained, and at a reasonable price?
3. Will they be used on your project? Will someone experienced with them be assigned to the project or will this be a training exercise for new kids?

B. What tools are important?

1. Test automation tools
2. Version control tools
3. Diagnostic tools
4. Test planning tools

C. *Efficient facilities for configuration testing?*
   1. How does the lab deal with standard operating environments? Can they quickly recreate clean systems configured with the right environments?
   2. Does the lab have facilities for fast setup and tear-down for printers, modems, video cards, sound cards, etc.? What do their workbenches look like?
   3. Do they have the equipment you need? Not "can they get" the equipment – do they actually *have* it, now?
   4. Do they have efficient processes for obtaining loaners from device manufacturers?
   5. If they have to borrow or rent the equipment, what access will you have to it? Will they be using this equipment for other companies' projects too? Will you be reimbursed for this use or are you paying for the full rental (as is, maybe, the other company?)

D. *Other lab issues*
   1. Does the vendor have enough room in the lab to accommodate the size of team that you will need?
   2. Who has access to the lab?
   3. Is there a separate "secret project" lab area? What security systems apply to it? Do you need it?
   4. If there is a separate "secret" lab, what does security *really* look like in the rest of the lab?
   5. Is lab equipment shared across projects or dedicated to specific projects?
      a. If shared, how does this affect security and efficiency?
      b. If dedicated, what does this cost you?

e. Use of 3rd party software to test your product

A. *If you will provide 3rd party software to the outsourcer, do you have the right to transfer or lend that software?*

B. *If the outsourcer will use 3rd party software on your product, does it have the right to use that software for that purpose?*

C. *If the outsourcer will use 3rd party software to test your product, what happens when you want to repeat some of its tests later? Can you retrieve test cases from archives and use them without the 3rd party tool? If not, does the outsourcer have the right to transfer a copy of this software to you?*

f. Experience with your type of application[11]

A. *Do they understand your market and your customers?*

B. *Do they understand your application, and how people will use it?*

C. *Do they understand the environment needed for this application (O/S, hardware, etc.?)*

D. *Do they have a supply of already-made test cases that are useful for this type of application?*

E. *Do they understand the regulatory requirements or issues for this application?*

F. *Do they understand the key competitive issues for this application?*

G. *Do they understand the key usability issues for this application?*

H. *Do they understand the key compatibility issues for this application?*

I. *Do they understand the key support issues for this application?*

g. Experience with your platform(s)

A. *Web apps need multi-level testing*

B. *Do they know server-side testing as well as client-side testing?*

---

[11] You might not have this experience either, but if you invest in training your own staff, you will have the experience next time. If you invest in the outsourcer to do the work, you won't get the experience.

C. *Give them some hypothetical bugs and ask them how they do follow-up testing to trace a bug to the server, the client, the database, the network, etc. Is their strategy sound?*

h. Experience with localization

A. *Does the outsourcer's staff speak the language to which you are translating the program?*
B. *Does the outsourcer know 2-byte language issues better than you do?*
C. *Does the outsourcer have experience in testing localizations for this language or this class of languages? (Issue: someone who has worked with translations into French or German might not understand the problems involved into translations into Chinese, Japanese, or Arabic.)*
D. *Does the outsourcer know enough about the target market to recognize when clip art or text, legal in the US, will be offensive or illegal in the target market?*

i. Willingness to work with you

A. *Some outsourcers insist on their view of The One True Way of software development. Is your religion the same as theirs?*
B. *Will they change their processes to accommodate you? Should they?*
C. *Will they change the relative time they spend on different types of work, in conformance with your preferences? Should they?*
D. *Will they report bugs in the way you want to receive the information?*
E. *Will they work with special intensity at critical times?*
F. *The vendor uses every change as an opportunity to charge more. They blame you for change rather than estimating in a way that anticipates change.*

j. References

A. *Has the outsourcer provided you with credible references?*
B. *At least some references should be to companies whose products and/or market are as similar to yours as possible. Press this point. If the outsourcer is reluctant to give you directly comparable references, then it may have no experience in your niche or it may have had a problem with a company you know. You'll want to look into this.*
C. *Have you checked them?*
    1. Describe overall project – where was the work done, why, how many staff on the project, was this the first time the client had worked with the outsourcer, etc.
    2. Timeliness of the work?
    3. Quality of the staff?
    4. Reasonability of attitude?
    5. Got the right equipment on time at the agreed price?
    6. Met expectations?
    7. Helped get the right product ship on time?
    8. Examples of disagreements, how were they handled?
    9. Did the outsourcer assign the people to the project that were expected (e.g. the ones the client was introduced to?)
    10. How well did the outsourcer know the market? How much effort did the outsourcer spend learning the market?
    11. How much ramp-up time, and at whose expense?
    12. Were there traffic jams in the outsourcer's lab and how were they resolved?
    13. Skill at automation?
    14. Skill at transferring knowledge to client's staff?
    15. Quality / thoroughness of testing documentation? (Appropriateness to what was requested, bid or ordered?)
    16. Status reporting? How done, how expensive, was client satisfied?
    17. Would client re-hire?

D. *Have you spent any time with the products the outsourcer tested? Were they any good? To what extent did the outsourcer play a role in improving the quality of these products?*
E. *What is the client's litigation history?*

6. Dispute resolution and risk management

a. I think it's essential to think these issues through up front. It's tempting to leave these hard discussions "for later" but the problem is that "later" never arises until there's a dispute. This is very different from other issues in the ongoing relationship, which are likely to get worked out, or at least worked on, many times.

b. Why this is so important:

A. *A good process lets you work with less tightly defined deals.*
B. *Typical formal dispute resolution processes (lawsuits, arbitrations) are unsatisfactory. They come too late and all they provide is compensation (money). They don't help move the project forward to completion.*

c. Ideas on resolution

A. *Resolve problems as early as possible.*
B. *Meet and confer at the liaison level.*
C. *Gradual, planned escalation through management levels.*
1. This sounds better than it works. Go up a few management levels and neither side might understand the issues or the technology. The result can be a deadlock in negotiations that is resolved in an unsatisfactory way.

D. *Mid-project mediation.*
E. *Mid-project mini-arbitration by an independent arbitrator / expert that you both trust.*
1. This is a session lasting 4-8 hours, in which both sides present the dispute to the expert, who issues a ruling within a week. The goal is to get a clear decision on the issue right away, and move forward. Even if the arbitrator makes the wrong decision, it is often better to move forward than to drag down the project by prolonged fighting over an issue that won't go away.

F. *Mid-project fact-finding by an independent expert that you both trust.*
1. The third party makes no decisions. She meets with you and the outsourcer, looks at documents, and writes a report that explains the controversy between you and the outsourcer, and that recommends a course of action.

G. *Agree in advance (in the contract) on the extent to which a mid-project arbitration will be binding. For relatively small disputes, the arbitration should always be binding. For very large disputes, different lawyers have different views on whether binding arbitration is a great idea or not.*
H. *If arbitration is non-binding, can the results be used / mentioned in a later court proceeding? Similarly for the mediator and for the fact-finding – should this be guaranteed to be confidential, in furtherance of open disclosure during these negotiations?*

d. Third-party helpers

A. *Agree up front about which people you will use as third-party helpers.*
B. *List a few alternatives, just in case your first choice isn't available.*
C. *Keep them out of all other aspects of your contract, to avoid conflicts of interest.*
D. *Require them to list their history with both sides and any conflicts of interest involving the project.*
E. *Use people who are credible on the technical level and who have reasonable business judgment.*
F. *For very large disputes (involving large sums of money), you might prefer a three-arbitrator panel instead of a single arbitrator.*

G. *Mediators don't have to be experts in the technical matters because they don't make the decisions. But they must be literate in the technology.*

e. What if the outsourcer completely screws up?

A. *Are you setting yourself up to be totally dependent on the outsourcer?*

B. *If you are convinced that this project will fail, are you trying to push everything to the outsourcer so that you can scapegoat the outsourcer when the failure occurs? In your company, will this tactic work?*

f. How will you manage the outsourcer's weaknesses?

A. *You can't safely assume that the outsourcer has no weaknesses. How will you discover and compensate for the weaknesses that it has?*

1. Do they know your market?
2. Do they understand your application?
3. Do they understand your customers?
4. Do they understand usability issues?
5. Do they understand your environment (O/S, hardware, etc.)?
6. Do they have a sufficiently stocked and organized equipment lab?
7. How junior are their staff?
8. How difficult is it for them to recruit new, experienced staff?
9. Who will do your glass box testing? (If the outsourcer can't do it, will you still be doing it in-house?)
10. Can the outsourcer measure its coverage?
11. How thoroughly does (can) the outsourcer investigate bugs before sending them to you?

B. *Break out the tasks and work with the outsourcer to appraise where they are strong and where they are weak. Plan to provide support for important issues in which they are weak.*

g. How will you deal with "unexpected" contingencies?

A. *What if the outsourcer ignores important issues, or just doesn't think they're important?*

B. *What if the outsourcer is testing too slowly? You seem to be getting much less of the lab's attention than you expected. You can fire an employee in this case. What can you do with the outsourcer?*

C. *What if the outsourcer fails to provide detailed test results? You get bug reports and verbal assurances, but not extensive test cases with a trail showing what cases were executed and what the results were?*

D. *What about the unexpected (if inevitable) last-minute shuffling of the feature set?*

E. *What if (some or all of) the software is too unstable to be tested?*

1. For how long?
2. What does the outsourcer's staff do in the interim?
3. Do you get back the same staff if there is a one-month dead time between testable versions? (Probably not? How does this compare to your in-house situation?)

F. *What about staff turnover? Is this planned for? Who pays to train replacement staff?*

G. *If the project runs far over schedule, what if the outsourcer has to transfer staff to a different project?*

H. *What if the project cancels before the planned completion? What do you owe the outsourcer?*

h. Backups

A. *Will the outsourcer make backups of its test plans, automation suites, bug reports, etc.? How often? What procedures will be used?*

i. Disaster recovery

A. *What plans exist if outsourcer has a disaster?*

1. Earthquakes, floods, etc. can cause the outsourcer to lose all of its data. Is there a remote backup site?
2. Are you getting reports and documents from the outsourcer so often that there is no need for a remote backup?

j. Force Majeure Clause

A. Is the outsourcer's failure to perform excused by a disaster (earthquake, etc.)?

B. What are your responsibilities to each other in this case?

C. Can you retain another outsourcer to fill in while this outsourcer recovers?

7. Intellectual property

a. The intellectual property rules can have a profound effect on the value of the deal for both parties. This is another area that is often ignored during contract formation, ignored during the project, until a dispute arises.

b. Who owns the work products of the testing?

A. Along with the (probably trade secret) bug reports, the work products include:

1. Test plans
   a. Some of this material might be generic, used or useful for developing templates for a wide range of products.
2. Test cases
   a. Some of these are specific to the software under test.
   b. Some of these might be operating system or environment compatibility tests or other generic tests.
   c. Some of these are borderline. For example, imagine testing a database client that communicates with the main database via SQL commands. The outsourcer has a standard set of SQL compatibility tests, and adds more to the set while testing your system. Should these be treated as specific to the software under test or generic?
3. Databases formed with sample test data
   a. Specific to your software or generic?
4. Automation scripts and programs
   a. Some of these are specific to the software under test
   b. Some of these are generic.
5. Tools developed for testing, such as load test tools
   a. Some of these are specific to the software under test
   b. Some of these are generic

B. Should the generic material belong to you or to the outsourcer?

1. The test lab wants to be able to reuse the technology that it develops.
2. You have to be able to reuse materials and technology that the test lab has developed for your product.
3. You want to be able to reuse materials that were developed for one product, when you test another product.
4. If the outsourcer keeps the intellectual property rights, you want a license (royalty free, perpetual) that allows you to reuse the materials for your product and that allows you to reuse the materials for some other project.

C. Should the material that is specific to your product belong to you or to the outsourcer?

1. It SHOULD belong to you, right? But If there is no agreement, your rights of use are limited. The test lab might well own the copyright on the test plan and its other work product, leaving YOU out of luck.

c. Intellectual property liability

A. What happens if you ask outsourcer to work on code in a way that exceeds your rights under the license? If the licensor sues the outsourcer, what is your liability?

B. Similarly, what if the licensor gives you access to 3rd party materials or uses them, outside of the scope of its license, and the licensor sues you. What is your liability?

C. Similarly for other intellectual property that one of you supplies to the other, exposing the other to liability. Who pays?

D. There are different considerations for:

1. Copyright liability
    a. The person who supplies the other with copyrighted materials should know that they are copyrighted and should normally be fully accountable.
2. Patent liability
    a. This is much more difficult. There can be patent infringement even if neither of you realizes that you are working with patented ideas.
    b. The fairer standard is to require each other to take reasonable care to not infringe.
    c. The issues are even tougher if the infringement is of a foreign patent.
    d. What about the case in which a third party infringes on a patent as a result of following your precise directions?
3. Trade secret liability
    a. The person who reveals the trade secret should be liable
4. Trademark infringement

C. Additional Standard Contractual Issues

1. These are typically perceived as more technical issues, but they still have an ongoing effect on the day-to-day relationship.

2. Confidentiality.

a. Define confidential information

A. How will each of you identify what information is confidential and what information is not?

B. If you label everything as confidential, nothing is confidential.

C. If you treat your sensitive material sloppily, it is hard to hold them accountable for leaks or disclosure.

b. The lab's responsibility to you

A. They sign nondisclosures on your products.

B. They sign nondisclosures on 3rd party products that you have, that you let them use (if you can let them use them at all).

C. They have effective security measures at their site, to protect your secrets.

1. The outsourcer might have a special locking room that they can reserve just for your project. You will pay extra for services like this.
2. The outsourcer's staff should have all signed nondisclosure agreements that bind them to respect nondisclosures that the outsourcer's management has signed with you.
    a. Don't assume this. Ask to see the standard nondisclosures and ask in your contract for assurance by the outsourcer that everyone who they involve with your product has signed or will sign this agreement.
    b. In any situation in which you have doubt, seriously consider asking the outsourcer for the specifics of that individual's nondisclosure agreement, or to asking the individual to sign an additional nondisclosure agreement with you directly.
3. The outsourcer's basic facility is locked, with secure locks, and ex-employees have no access.
4. People (non-employees) can't just walk in or out during working hours.
5. Network facilities are secure. Hackers on the net can't just break in electronically and gain access to your documents.

D. *If their staff are assigned to other projects as well as yours, can you avoid accidental disclosure of your secrets?*
E. *If your product is being tested in a shared lab facility (owned by the outsourcer, but shared across projects), will people from other project teams see proprietary material of yours while they work in the lab? If so, what confidentiality rules govern them?*
F. *Can any of the outsourcer's staff write reviews of products tested by the outsourcer without your approval?*
G. *Do they reveal other peoples' secrets to you?*

c. Your responsibility to the lab

A. *Should you be restricted from showing other labs/outsourcers who work with you the test planning materials or test cases that you got from this outsourcer?*
B. *What about nondisclosures involving their use of their proprietary tools on your projects?*
C. *What about nondisclosures involving their use of proprietary third party tools on your projects?*

d. Survival of the nondisclosure agreement

A. *Your duty to protect each other's secrets should survive the termination of the contract, and should be independent of the reason for termination of the contract.*
B. *Either of you can be compelled by court to testify about the confidential business-related issues or facts. However, the contract can specify that each of you will notify the other as soon as they receive formal or informal notice that they will or may have to reveal the other's secrets, and that they will cooperate with your counsel in using lawful methods to protect your secrets or to limit the scope of their disclosure.*

e. Electronic communication

A. *Will your electronic communications be encrypted? Protected in some other way?*
B. *If you are sharing data at a web site or at a "private" BBS or a private forum on an online service provider's system, what arrangements guarantee the security of the site?*

3. Non-competition clauses.

a. In general, I think these are inappropriate. The smaller the outsourcer (or the smaller you are), the more likely it is that the bigger guy will try to foist this on the littler guy. In general, these are negotiable, even in contracts that are initially presented as "non-negotiable."

A. *The courts are likely to enforce noncompetes between businesses.*
B. *The courts are less likely to enforce noncompetes against individuals but the general trend over the past decade has been in favor of business against individuals.*

b. They promise not to provide services to your competitors for X months.

A. *This is a dangerous clause for the outsourcer. Except under very special circumstances, they will not (or should not) sign it.*

c. They promise to take all the business you can give them.

A. *Do you have a right to place further projects at their site, even if this will make them too busy to take on other clients?*

d. You promise not to work with other outsourcers.

A. *This is a dangerous clause for you. What are its benefits?*
B. *Does the outsourcer have a right of first refusal on your next projects?*

e. You each promise not to provide any type of services to the others' clients.

A. *If you and they both engage in consulting, a clause like this might appear in the contract. It appears in some contracts as a matter of boilerplate. How does it limit your options? In general, I think that these create unknown limits and so I am reluctant to sign them?*

f. You each promise not to create new products or offer new services that would compete with the services currently offered by the other.

A. *This is showing up in some boilerplate contracts. Unless there is a strong business reason for it that applies directly to the relationship at hand, I think it is unreasonable.*

4. Employment issues.

a. Non-recruitment clauses

A. *You promise not to recruit their staff for X months and they promise not to recruit yours.*

B. *What about promises that you will not recruit former employees of each other? (These are increasingly often requested, I think they are unreasonable.)*

b. They act as an employment agency

A. *You promise to pay them for staff of theirs that you recruit. Does the amount depend on the number of months that their staff member has worked on your projects? (Some companies charge much less if you recruit staff who've already been working on your premises for several months.)*

c. They take over some of your staff

1. Your agreement may anticipate that you will terminate employment of some of your staff and that the outsourcer will hire these people. This is a complex situation, that must be worked out with care and specificity.

5. Conflict of obligations or interests

a. No conflicting contractual obligations

A. *This agreement and the transactions contemplated herein do not conflict in any material manner with any other agreements or transactions to which either party is a party or by which either party may be bound.*

b. No conflicting resource obligations

A. *The outsourcer has or can get enough staff and facilities to do the job properly. No one else who might compete with you for resources has a higher priority call on the outsourcer's staff and facilities.*

c. No unknown conflict of interest

A. *Is the outsourcer a shareholder in your competitor?*

B. *Is the outsourcer subject to a long-term contract with your competitor, which makes the outsourcer loyal to the competitor?*

6. Liability for quality of the work

a. Basic Liability of the Outsourcer to You

A. *Can you sue the outsourcer for bad work?*

1. How do you distinguish between bad software and bad testing?
2. Did you give the outsourcer enough time to do the job properly? (If you're working the outsourcer on an hourly basis, it's up to you to give the outsourcer enough time (though it's up to the outsourcer to ask for it? On the other hand, if you are working on a fixed-priced bid, then it is the outsourcer's responsibility to allocate sufficient time.)
3. What is your definition of "bad work?" Negligence? Gross negligence?
4. To what degree did you set the objectives, standards, and practices in use by the outsourcer?

B. *Can your customer sue the outsourcer for bad work?*

1. What does your contract say? Does the outsourcer promise to advise you of the reliability or safety of the product? Is it clear in the contract that you will ship the product in reliance on the advice of the outsourcer?
2. What does the outsourcer say to the public? Will customers buy the product in reliance on the reputation of the outsourcer?
3. Did the outsourcer certify the product (perhaps for a government agency) as having been appropriately and sufficiently tested?

b. Warranties

A. Expect to see disclaimers of implied warranties of merchantability, fitness, and any other implied warranties. These warranties are unusual in service contracts.

B. If the contract is silent, then the implied warranties might be in force under UCITA.

C. What warranties <u>will</u> the outsourcer make?

1. Intellectual property.
2. Promises that it has actually made regarding services, products, personnel, effort.
3. It is fair and reasonable to ask for warranties of workmanlike performance.
4. Because of the licensing / malpractice problems, an outsourcer would be a fool to agree to provide services that meet "highest professional standards" or "professional levels." We don't know what those standards are, but juries in places like Texas will soon have to invent them.

c. Indemnity clauses.

A. Indemnification for quality-related issues is rare and dangerous to the outsourcer.

1. It is absolutely unfair to demand such a warranty if you control the quality of the product.
2. It is appropriate for the indemnifying outsourcer to demand several additional passes at the software, and the budget to do deeper work.

B. Indemnification that is not quality-related:

1. Liability for torts committed by outsourcer's staff on your premises.
2. Liability for tax-related problems caused or triggered by the outsourcer.
3. Liability for outsourcer's unauthorized use of 3rd party intellectual property.
4. Liability for your unauthorized use of 3rd party intellectual property, given to you (without warning and without identification as 3rd party material) by the outsourcer.

C. Common issues

1. The indemnifier needs notice of all claims, and the right to co-defend them.
2. The indemnifier needs the right to settle all claims.
3. The indemnifier needs the right to approve all settlements.
4. Will the indemnifier pay for all claims (like insurance) or only in the event that it has actually breached one of its warranties or actually committed a tort?

d. Liability insurance.

A. There are lots of different types of policies. You have to decide which ones are relevant and essential. It's worth asking, for each of these, whether the outsourcer carries the insurance. If so, ask to be named as an insured. But if not, then you should think carefully before deciding that the outsourcer MUST go and purchase it.

1. Errors & Omissions / Professional liability
2. General Commercial Liability (probably doesn't include coverage for bugs, inadequate testing and general bad workmanship.)
3. Automobile liability
4. Workers' compensation
5. Employee dishonesty (bonding)
6. Key employee

7. **Other liability clauses**
   a. **Damage limitation clauses – subject matter**
      A. *for breach of contract (services not performed or warranties not met)*
      B. *for trade secret leaks*
      C. *for intellectual property infringement*
      D. *for negligence unrelated to the quality of the work (such as accidents on your premises or harassment of your or the outsourcer's staff)*
   b. **Damage limitations**
      A. *Consequential damages.*
      B. *Consequential damages for only certain specified types of losses, such as reimbursement of your costs in a copyright infringement caseby a 3rd party against you, in which the outsourcer was the real infringer.*
      C. *Incidental expenses*
      D. *Refund for services rendered, no other damages..*
      E. *No damages payable, only redoing of bad work.*
      F. *Limited to all or some percentage of the amount you paid.*
8. **Publicity**
   a. **What the outsourcer can say about the contract**
      A. *You might wish to limit the outsourcer's right to use your name in connection with its advertising or marketing.*
   b. **What you can say about the contract**
      A. *The outsourcer might wish to limit your right to use its name in connection with your advertising or marketing. (For example, it might feel this is necessary to limit its potential liability to your customers.)*
9. **Term and termination**
   a. **Agreed term (duration) of the contract**
      A. *This one project?*
      B. *All projects for X amount of time?*
      C. *The period of work (at your site) of listed individuals?*
      D. *Open-ended agreement for continuing services, probably with a cap on the total amount of money to be spent?*
      E. *Will you give the outsourcer a guaranteed minimum amount of business?*
   b. **Your ability to terminate before the end of the term.**
      A. *Terminate at will on reasonable notice.*
         1. Notice period?
         2. No statement of cause required?
         3. Is there an early-termination fee?
         4. Can you ban a specific individual of the outsourcer's staff from your premises without specifying a cause? (Just say "we don't like him, he goes away.")
      B. *Terminate for cause.*
         1. Immediate termination.
         2. What is sufficient cause? (The following may or may not be.)
            a. Laggardly work / missed deadlines?
            b. Continuing (despite protests) inadequacy in status reporting?
            c. Failure to provide agreed deliverables?
            d. Agreed staff members not working (or not working full time) on the project?
            e. Theft
            f. Intellectual property problems
               A. Trade secret leaks
               B. Unauthorized use of software, could expose you to liability
               C. Unauthorized copying or distribution of your software
            g. Non-competition

A. In violation of agreement
B. Even though there is no agreement, the lab just took on a project with your competitor that directly competes with this product. Is this a problem for you or not? Why?

h. Harassment or other torts against your staff or property.

c. Outsourcer's ability to terminate before the end of term

A. Terminate at will on reasonable notice

1. Notice period?
2. No statement of cause required?
3. Is there an early-termination fee?
4. Can the outsourcer pull key staff members off your project without justifying the decision? (For example, if one of the outsourcer's staff feels sexually harassed by your staff, can the outsourcer simply pull that person off the project or do they have to let you know?)

B. Terminate for cause

1. What if your project drags on way longer than the agreed term?
   a. Can the outsourcer terminate your contract in order to free up the time to work on another client's projects?
   b. Is this a termination for cause?
2. Does termination require notice?
3. What is sufficient cause?
   a. Non-payment
   b. Intermittent delivery of new versions, causing excessive ramp-up and slow-down times, wreaks havoc with outsourcer's staffing schedules?
   c. Continuing (despite protests) inadequacy in status reporting (by you to outsourcer, e.g. what has changed in the software)?
   d. Failure to provide agreed deliverables?
   e. Agreed staff members not working (or not working full time) on the project?
   f. Theft
   g. Intellectual property problems
      A. Trade secret leaks
      B. Unauthorized use of software, could expose outsourcer to liability.
      C. Unauthorized copying or distribution of outsourcer's software or materials.
   h. Non-competition
      A. In violation of agreement
   i. Harassment or other torts against outsourcer's staff or property.

d. Notification of termination

A. Who is authorized to terminate the contract?

B. Who does the authorized party have to notify? (Who do you send the letter to, who does the outsourcer send the letter to?)

C. Notification must be in writing? E-mail OK? Telephone? Certified mail?

e. Termination Support

A. Should the contract obligate the outsourcer to provide termination and transition support. (This additional support will usually be at your expense.)?:

1. Outsourcer accepts a duty to cooperate with the next outsourcer and to do all reasonable things necessary to make the transition as easy and inexpensive for you as possible. This includes transferring documents, test plans, source code for all test suites, test cases and sample data, bug reports, etc.
2. Outsourcer agrees to prepare additional documentation to provide to you or to its successor (additional test planning material, documentation of test suites, etc.)
3. Outsourcer agrees to make its staff available to you or to the next outsourcer for purposes of explaining its materials and its approach to testing your software, writing automated scripts, etc.

4. Outsourcer agrees to provide maintenance for its automated test suites or to train your staff (next outsourcer's staff) in the information needed for maintenance.
5. If the bug tracking is done on the outsourcer's database, outsourcer agrees to help port this data to a file format that you or the next outsourcer can use.

B. This type of agreement is particularly desirable if the outsourcer is going to take over all of your testing. It also gives you some reassurance / help if the outsourcer can terminate mid-project, or if the outsourcer is testing a base version of a product that might be updated or modified, with future testing to be done by you or by a different outsourcer.

C. Should contract provide for best efforts or reasonable efforts? Probably depends on the extent of your reliance on the outsourcer?

f. Access to source code

A. If the outsourcer is developing automated tests, will you have access to the source code in the event of a dispute or of termination?

B. Is the outsourcer required to archive the source code with a third party (escrow)?

1. If there is escrow, what measures is the escrow agent to take to ensure that the archived code is the actual code in use.

g. Termination during a dispute

A. Should the contract obligate the outsourcer to provide you data and cooperate in your transition during a dispute (especially a dispute in which you are not paying the outsourcer's fee)?

D. MANAGING THE ONGOING RELATIONSHIP

1. Who has final authority on testing priorities?

a. Who determines how extensive the test planning documents are?
b. Who decides which areas of the program must be tested next?
c. Who decides what features and issues are of the greatest importance to your customers? If this is the outsourcer's staff, where did they get that knowledge?
d. Who determines what equipment is used for the test?
e. Who determines what testing tools are used?
f. Are the priorities safer in the outsourcer's hands? (Do the politics of your company make it impossible for you to do thorough testing? Can the outsourcer take a hard-nosed position?)

2. How independent is the outsourcer?

a. Is independence *required* by the contract? (The customer might require an "independent" test.)
b. Even if the outsourcer is officially playing an "independent" role, there are several ways that companies pressure outsourcers to be lenient. For example, if your company has a continuing relationship (many projects) with the lab, then if the lab is too feisty, you will take your business elsewhere.
c. You might closely manage the outsourcer. A test lab need not be "independent."
d. Consider having your staff train the outsourcer's staff. This gives your testers a break, and it can build team unity.

3. How will you supervise the effort?

a. You must supervise the effort.

A. Replicate all bugs reported by the outsourcer.

B. Look for more serious consequences of these bugs.

C. Audit the test plan. Find an organized method to test it for holes. Don't just "review" it.

D. Read the programmers' responses to bug reports. Look for communication problems.

E. Monitor progress against the project schedule.

F. Monitor testing progress against the apparent maturity of the rest of the project. (Is testing going faster or slower than documentation, tech support, etc.)

G. What status reports will you receive?

H. If the outsourcer will give you an automated test suite as a delivery, how will you know whether your staff will have enough information to be able to reuse the suite next year?

b. Liaison with the outsourcer?

A. You need a liaison.

1. This person does the first-level dispute resolution with the outsourcer.
2. Primary reviewer of the quality of the outsourcer's work.
3. Identified person for your staff to bring problems to (e.g. if your programmers don't like the bug reports, they complain to the liaison.)
4. Identified person for the outsourcer to bring problems (their problems with you) to.
5. Most knowledgeable person about the outsourcer's business and technical practices. Can provide insight.

B. Typical tasks of the liaison

1. Review all test plans.
2. Reports on testing project progress to you and your management.
3. Meets with outsourcer, conveys information about schedule, timing and content of upcoming deliveries, as well as raising problems and frustrations.
4. Approves minor changes in the project plan within a defined scope of authority.
5. Meets with the outsourcer, carries back information on the outsourcer's difficulties and frustrations.
6. Gets clarity on what tasks the outsourcer is not going to do. (Including tasks that you would normally do but that the contract does not require the outsourcer to do.) Flags these in memos to management and recommends ways to get these tasks taken care of.
7. Source of all material sent to the outsourcer.
8. Archives all releases sent to the outsourcer.
9. Receives all material from the outsourcer.
10. Archives all materials received from the outsourcer.
11. Replicate and extend each bug.
12. Review all bug reports. Look for communications issues during the bugs' lifecycles. Follows up on unresolved queries regarding reproducibility, significance, etc.
13. Monitor your staff's reasonability in handling and commenting on bug reports. (If people treat the outsourcer without respect, in ways that will waste the outsourcer's expensive time, this is an issue to be dealt with in-house.)
14. Run independent tests to check effectiveness of testing.
15. Audit test coverage.
    a. Check whether the test plan would capture hypothetical bugs.
    b. Check whether coverage measurement is actually being done (if it should be).
    c. Use other techniques as appropriate to check the actual extent and depth of testing.
16. Control point for all configuration testing: approves what configurations will be tested; approves reduction in configuration test plan.
17. Carries / manages tech support requests about the format of material that shows the results of configuration testing or of other types of tests that have tech support implications.
18. The liaison should know every tester.

C. Skills of a good liaison.

1. Extensive testing experience.
2. Understands test management.
3. Understands testing-related economics and overall project economics.
4. Understands the risks of test automation.
5. Understands the risks of excess project documentation / bureaucracy.
6. Understands enough about software development to understand the technical debates involving the product.
7. Diplomatic, but firm. Able to explain your company's needs and to articulate your company's disappointments.
8. Charismatic enough that the testers are comfortable working with, and opening up to, the liaison.
9. Level-headed. Calm in arguments. Able to deal with pressure and whining from all sides.
10. Good listener.
11. Knows your market (or willing, able to find out).
12. Methodical and attentive to detail.
13. Able to keep secrets.
14. Loyal to your company. Ideally, has an attractive, long term career path in your company, so it is worth actively protecting your company's interests.

D. Your liaison needs sufficient time to do the job properly.

1. Has to come up to speed on the program.
2. Has to understand the market.
3. Needs time to review / reproduce the bug reports.
4. Needs time to review / audit the test planning materials.
5. Needs time to review the progress reports.

E. Where should the liaison work?

1. Their site
   a. S/he sees what's going on.
   b. S/he can spot check the tests in progress.
   c. S/he builds personal relationships with outsourcer's staff and can coach them.
   d. But s/he may come to identify with them, at your expense.
2. Their liaison works at your site
   a. S/he sees (and can more easily come to understand) your frustration with their work. S/he might come to empathize with your point of view.
   b. S/he can get technical information from your staff quickly.
   c. S/he can resolve many communication problems with your staff more quickly.
   d. S/he can dynamically readjust the testing staff's priorities, based on the actual project progress. The more information s/he has, the more efficiently the outsourcer can manage its schedule.
   e. But s/he sees your dirty laundry.
   f. But s/he knows where your company's pressure points are; might be too effective a negotiator against you in a dispute.

F. Do you have a liaison backup in case of turnover?

c. Outsourcer's liaison with you?

A. Name the person, or the managerial level of the person

B. Name the duties of the liaison

C. What is the negotiating authority of the liaison? What can this person agree to without needing approval from more senior management?

4. Where does the outsourcer's staff work and who manages them?

a. Management structure

A. Fully independent group working at the outsourcer's lab?

B. Working at the outsourcer's lab, but with a supervisor of yours at their site?

C. Independent team working at your site?

D. Team working at your site, under one of your supervisors?

E. Team working at your site, as members of a larger test group working on the same project?

b. Reasons you might want the additional resources locally, at your site

A. Local systems issues:

1. You have specialized equipment or configurations onsite.
2. Your client (custom software) has provided you with a specialized system or system software, as a unique resource, at your site.
3. You're not set up for the outsourcer (or anyone else) to access your system, or they won't use it remotely.
4. You have a license for a specialized tool, which the outsourcer doesn't have. Their staff has to come to your place to use the tool.
5. The outsourcer needs regular access to your local source or source control system and these can't be echoed offsite or put on your website.
6. Can they recreate your web server at their site? Do they rely on your server? Can they access it from their lab? If something goes wrong, how will they diagnose whether the problem is at the client side, the hookup, the server, etc.?
7. The outsourcer can't fit enough more people into its facilities. They have to work at your site.

B. Preference for face to face communication because:

1. You have no specifications.
2. You have an inadequate written test plan.
3. Unless the outsourcer's staff is onsite, there won't be any communication between them and your programming staff.
4. Some development groups respect local testers more than offsite people.
5. Some groups don't trust offsite workers.
6. Some groups won't take criticism from faceless third parties. If they can't discuss a bug report, they won't take it seriously.
7. Local folklore: your staff "always" test for some things but don't write them down.

C. You want to exercise local control.

1. Maximizes your ability to manage the outsourcer.
2. You can manage their allocation of time.
3. You can shift them from issue to issue depending on the other chaotic shifts in development schedule and progress on the project.
4. You can more easily organize and reset priorities.
5. You might want to give the outsourcer more local political influence.

D. You want to see the work.

1. Need to determine if this outsourcer's staff have a clue.
2. You can audit their work by watching it.
3. You can see that the outsourcer's staff is working full-time on your project.
4. You can see that the outsourcer's staff is probably also not working on your competitor's project.

E. You want a skill transfer from outsourcer's staff to yours.

1. They show your what they do, while both groups work side by side.
2. Get "neutral" view of your process (to make a case for changes?)

F. Security requirements

1. Need for day-to-day identification of all people who have access to your materials.
2. Need a daily record of access, based on positive ID (biometrics, fingerprints, whatever).
3. You want to ensure that no one has access to your product who is on the outsourcer's staff but not on your project.
4. You have a secure lab and want all testing and development done there.

G. Recruiting

1. The outsourcer has to recruit to meet your staffing needs, but the labor market is better in your city than in the outsourcer's lab's city.

H. Their physical plant is bad.

1. Is their room or work environment comfortable? Put yourself in the place of the tester and ask yourself if you could work there. If you do think so, don't expect their staff to do well there.
2. Is the environment noisy? Distracting? (Is yours better?)

I. Time zone issues

1. Is it convenient to talk with them across the time zones?
2. If they are relying on remote access to your equipment, does a time zone difference complicate this?
3. If the system goes down in your off hours, how long will it take to restore it so that the remote testers can use it?

c. Reasons you might want the resources at the outsourcer's site.

A. Your location is inconvenient

1. There's no space for more people at your site.
2. The space that you would provide to the team is too crowded, uncomfortable, or noisy.
3. The overhead costs are much lower at the outsourcer's site (city, country).
4. They have specialists onsite, such as language experts, who have to be consulted intermittently.
5. There are experienced people at the outsourcer's site who won't travel.
6. The outsourcer will give you a discount if you let them do the work at their site.

B. Local systems issues:

1. The outsourcer has specialized equipment or configurations at its site.
2. The outsourcer has non-portable software that you don't have.
3. Infrastructure of outsourcer – hierarchy of leads, processes, etc.
4. They have more bandwidth available for a specific task than you.

C. Communication issues

1. You want the staff on your project coached by the outsourcer's senior staff.
2. You want to contract the services of a talented coach who works at the outsourcer's site.
3. Some companies respect opinions of external testers more than internal staff.
4. The outsourcer can give you more or better (more detailed) status reporting and invoicing if the work and their supervision and tracking of the work are done at their site.

D. You don't want to exercise local control.

1. You don't want to manage the details of the project.
2. You think this project will fail and you want it to fail far, far away from you.
3. There's no glory in "point" releases (in-line updates) and so you want to minimize your investment.
4. This is legacy testing, you find it really boring and you want it out of your hair.
5. The outsourcer can give you better staff utilization reports when work is done at their site, using their time tracking systems.
6. The outsourcer's metrics are local to them, independent of your approach, and you want to be able to present them that way.
7. The outsourcer is doing independent validation and verification and must be isolated from your company.
8. If the outsourcer is seen as independent of you, then their opinions will be treated as outside opinions. These might be taken more seriously than in-house opinions.

9. The outsourcer might have greater political power based on their independence and remoteness. (Familiarity breads dismissal.)

E. On a short fuse schedule, you can take advantage of time difference issues.

F. You want a skill transfer from outsourcer's staff to yours.

1. You think the best way to train your staff is to send them to the outsourcer's site.
2. This helps you and your staff learn about the outsourcer's documentation and deliverables.
3. The travel expenses are cheaper if you send your (few) staff to the outsourcer's site than if you try to import the many people from their site.

G. Security requirements

1. The outsourcer's staff can't look over your other projects.

H. Recruiting

1. It's easier to recruit in the outsourcer's city.
2. The outsourcers claim that there is a broader experience base at their city than at yours.
3. You don't want to recruit in a way that reveals that you are doing or involved in the recruiting.

d. You may need to work at both sites

A. For example, maybe you have the server or mainframe at your site and the outsourcer does the client-side testing at theirs.

1. Who does the server side testing? You or them?

5. What status reporting do you expect?

a. Administrative overhead costs money. How much are you willing to pay for status reporting?

A. You might want more detail than you generate in-house because it's harder to see the progress and to learn about it in other informal ways.

B. You might want more detail as a matter of good contract management.

C. But every additional level of detail in status reporting costs more money.

b. What items do you want reported periodically?

A. (What period? Weekly? Per cycle of testing?)

B. How many new bugs found.

C. Number of bugs closed as not-fixed (e.g. "Works as designed", "not a bug", "deferred", "not reproducible").

D. Totals of found / fixed / closed as not-fixed.

E. Number of new test cases created.

F. Number of new test cases executed.

G. Percent coverage (against what coverage criteria) achieved.

H. Detailed progress shown (maybe using a project manager) against a detailed task breakdown.

6. Deliverables

a. Shipping and handling and packing and marking instructions

A. This causes more disputes than you'd expect. Where should documents and hardware go? (Same place, different places?) What special rules are there for packing equipment (Such as, ship hardware in its original packing?)

b. What deliverables are you supposed to provide?

A. The software, and updates (how often?).

B. Overall project schedule (software / doc / etc. delivery milestones and dates).

C. Ongoing progress reporting against the schedule.

D. Product documentation?

1. User docs.
2. Requirements.

3. Specifications.
4. Regulations that dictate features or other requirements of the finished product.

E. Hardware for testing? (Keep a list of anything you lend the outsourcer.)
F. Tools? (Keep a list of anything you lend the outsourcer.)
G. Other software, for compatibility testing or for benchmark testing? (Keep a list of anything you lend the outsourcer.)
H. Test cases?
I. Test data?
J. Bug reports (whenever you find bugs that they missed?)
K. Standard compatibility / compliance test suites?
L. Copies of relevant regulations?
M. Market profile information?
N. History of customer complaints with this product or its competitors?
O. People?
P. Office space or lab space for the outsourcer's staff?
Q. At the end of the project, what do you have to give to the outsourcer?
1. Hardware that the outsourcer loaned you / put in your lab?
2. Tools?
3. Other software
4. (You want to have a written list of everything that comes to you, to avoid disputes at end of project.)
5. End of project evaluation?

c. What deliverables is the outsourcer to provide?

A. Bug reports.
B. Project or product metrics?
C. End-of-project appraisal of the software?
D. Project post-release review?
E. Overall testing project plan and schedule.
F. Detailed test plan.
1. Will you use a standard, such as IEEE 829, as a shopping list to explain what should and should not be in the test plan?
2. Do you expect this 100% done up front or 10% done up-front with 90% evolution while testing, or something in between?
3. How detailed?
4. Should some aspects be more detailed than others?

G. Reusable test cases?
H. Automated test cases?
1. Documentation?
   a. Of individual cases?
   b. Of expected results?
   c. Of coverage achieved by the test suite?
   d. Will this be maintainable by your staff next year?
2. Test harness?
3. Training in their use?

I. Hardware for testing? (Did you contract with them because they have a big printer lab? OK, do they have the printers you need tested?)
J. Market data?
K. Benchmark data from competitive products?
L. Customer complaint information for this class of product?
M. People?
N. Equipment and software compatibility reports in a form suitable for use by your support staff?

O. *Office and lab space for these people?*
P. *Standard test data (such as 3rd party-published compatibility test suites?)*
Q. *At the end of the project, what does the outsourcer have to give to you?*
    1. Hardware that you loaned the outsourcer.
    2. All copies of your software, manuals, etc. (To preserve confidentiality, how much of your in-development material should the outsourcer be allowed to keep?)
    3. Tools.
    4. Other software.
    5. (You want to have a written, signed off, list of everything that you have loaned the outsourcer to avoid disputes at end of project. You should also have ID tags on all of your equipment, software, books, tools, and anything else that you want back from the outsourcer. It's not that the outsourcer is dishonest. The problem is that they handle volumes of this stuff and it's too easy for your stuff to get mixed in with somebody else's stuff.)
    6. Any software that the outsourcer wrote to enable testing of your product.
    7. Final archived set of test cases and results.
    8. Full list of all configurations tested.
    9. All test files, test plans, test scripts, with documentation to help you figure out what is and is not covered in these tests, and to help you maintain the test suites in the future.
    10. End of project evaluation.

d. What stuff is coming from 3rd parties?

A. *Hardware on loan?*
B. *Software on loan?*
C. *Specifications or requirements?*
D. *3rd party auditors or other staff?*
E. *Training, course material, coaching on specific technical issues.*
F. *Who will manage the list of 3rd party material?*
    1. Making initial arrangements for loaners or other materials.
    2. Returning what needs to go back.
    3. Nagging / begging for items that haven't been received.
G. *Who is responsible to the 3rd party to get their equipment, material back to them?*
    1. If the outsourcer is responsible, but fails to return the equipment, can the 3rd party come back at you?

e. Change Control

A. *If you provide new versions of the software, what are your responsibilities to document changes?*
B. *What are your responsibilities to limit changes?*
C. *Your responsibilities might be more different, and more seriously enforced, in a fixed-price bid. The outsourcer doesn't want to agree to $X for program Y and then be given program Z to test. Or a slow, expensive (unexpected by the outsourcer) transition from Y to Z.*
D. *To achieve change control, what level of access do you have to give the outsourcer to your corporate decision making process (do outsourcer representatives sit on your change control committee?)*
E. *What degree of access do you have to give the outsourcer to your internal data systems (e.g. source control system)?*
F. *How much extra documentation will you generate (if any) to achieve the change control requirements, and who will do this work (will your programmers agree to it?) How much will this cost?*

7. Approvals

a. State in the contract, who approves which piece of work.

b. What is the time frame for acceptance review of each piece of work.
c. What is the process for inspecting each piece of work submitted.

8. What is the bug reporting / resolution process?
a. Whose bug tracking system is used? Why?
A. If they don't enter bugs into your system, how much work will it take you to get the reports into your system?
b. How are bugs transmitted from them to you, and responses back?
c. What is the bug lifecycle from open to closure?
d. Does the bug lifecycle include as many checks and safeguards against shipping bad known bugs as your in-house systems?
e. Who prioritizes bug fixes?
f. Does the outsourcer report design issues or only coding errors? Who decides whether the outsourcer should or should not report these?
g. Should the outsourcer report nonreproducible failures? If so, how much troubleshooting should they do first?
h. In the event of a nonreproducible failure, should they call into your programmers to ask for guidance or hints?
i. Who decides whether a bug is acceptable in the field? If this is the outsourcer's decision? What if you disagree?
j. When you find new bugs, do you report / copy them to the outsourcer?

9. Measurement of Performance
a. What performance standards are there for quality of work or product?
A. Verbal assurances of expertise and other verbal promises are meaningless, because there will be an integration clause in the contract that says that anything not directly promised in the contract is not part of the deal.
B. Should the contract include a "best efforts" clause that obligates the lab to use its best efforts to meet the schedule, to automate the tests fully or to achieve some other goal(s)?
C. Will you use a standardless review as the requirement for your signoff? (Such as a simple reading-based review of the test plan?) (What if you and the outsourcer disagree about whether the test plan is up to par?)
D. Will you use a standardless independent audit of the test planning and testing?
E. Will you measure for conformance to industry or internal standards?
F. Will you use a coverage criterion?
G. Will you create specific performance measures? Beware what you measure, because that's what the project will be optimized for.
1. Testers expected to report 5 bugs per day? (I don't recommend bug-count standards.)
2. Not more than X% bugs irreproducible.
3. Not more than X% bugs come back with queries for more info about the bug.
4. Response time to clarify or supplement a bug report is within Y days.
5. Time needed between receipt of a new version of the software and completion of the acceptance-into-testing (smoke) tests.
6. Availability of the outsourcer's testing staff to answer questions, meet with your staff, etc.
7. How many test cases are coded per day?
    a. Can you subdivide this in terms of complexity, power of the test cases?
    b. This is a dangerous measure, you might get lots of poor tests. How do you avoid getting 1000 trivial test cases because all you're measuring is the absolute number, independent of quality?
8. How many devices are tested per week?
9. How many environments (O/S plus system hardware configuration) are tested per week?

b. Can you create a bonus system for high quality work?

A. *For example, bonus based on how many not-discovered-in-testing bugs were found in the field? (Fewer is better.)*
    1. Should bugs that the outsourcer wrote up as minor in the database be treated as equivalent to non-discovered if they turn out ot be serious in the field?
B. *As another example, should you give a bonus for passing a tough audit?*

E. TESTABILITY ISSUES

1. **This section is largely redundant with the rest of the outline, but it raises related issues in one place. The problem is that if your software is hard to test, then you will either spend more on outsourced testing or you will get less testing.**
    a. **The problem is worse for outsourced testing because their staff probably don't have as much access to your programmers as your staff does.**
    b. **Here are notes on some of the issues that are sometimes mentioned in discussions of testability.**
    c. **Many products and development processes will not provide for everything in this list.**
2. **Background info for the tester**
    a. **Requirements, specifications, descriptions of development intentions.**
    b. **Lists of all variables, all error messages, all reports produced by the software, all dialogs, etc. (Boundary conditions and descriptions of the relationships among the variables would be nice too; the more you give the outsourcer, the more efficiently they can test.)**
    c. **Competitive products: issues in the marketplace involving features, reliability, usability, and compatibility with other key products or devices.**
    d. **Externally imposed requirements, such as regulations and standards.**
    e. **Hazards associated with products of this type. What kinds of risks do these products pose to customers or to their property?**
    f. **Bugs typical of products of this type or typical of this platform. (If your product or platform is specialized, you will know more about this than the outsourcer.)**
    g. **Copies of previous test plans and test cases.**
3. **Visibility and control**
    a. **One-to-one correspondence between problems detected by the code and error messages. Different problems don't result in the same message.**
    b. **APIs or other mechanisms that allow testers write test code that bypasses the user interface.**
    c. **Error/event log mechanism to give tester visibility into internal error detection and handling. Or, tester can query internal states and variables.**
    d. **Ability to force values into calculated variables, in order to check error handling.**
    e. **All factors affecting output are visible.**
    f. **The program (or an appropriate, connected tool) can generate a log file that shows the execution history.**
    g. **The program will automatically detect and report internal errors.**
4. **Configuration management and change control**
    a. **Unique version numbering for every version of the software sent to the outsourcer.**
    b. **Source control and archiving to make it easy to recreate any version sent to the oursourcer.**
    c. **Specific but tester-comprehensible notes that document changes made to the code since the last version sent to the outsourcer.**
    d. **To what extent will changes made to the software cause rework of the outsourcer's test suites and test documentation?**

A. *(Obviously, life would be easier for the outsourcer if you delivered absolutely final product and made no changes, or none to any visible function. Beware of wasting the outsourcer's time, but be wary of temptations to let the outsourcer tail wag the developer dog.)*

5. **Comparable equipment**
   a. **Can you recreate the test setup in the outsourcer's lab? Often not. (That's why you hired the outsourcer–they have this big lab . . .) How are you going to manage the problem that they can recreate a bug in their lab that you can't recreate in yours? For example, what diagnostic tools will you provide with the software?**
6. **Bugginess**
   a. **The more buggy the software, the more time the outsourcer will spend reporting and working around bugs.**
      A. *Software developers that are contractually or legally obliged to send software to an independent software test agency will often hold their software, not releasing it to the outsourcer until it is quite stable.*
   b. **Bugs that block testing must be fixed quickly.**
      A. *Someone on your staff must take ownership of this problem.*
7. **Automation support**
   a. **In a Windows product, is your company using custom controls? How do these interact with the automation tools that you use?**
8. **Support for early testing**
   a. **In a collaborative relationship with the outsourcer, there's much to be gained by beginning testing as early as possible. You have fewer problems when you test incrementally, as the code is written, instead of waiting for a big bang (all code delivered at once, none of it works, kaboom—everything crashes together, no one can figure out what particular bug led to these particular symptoms.)**

F. OTHER STANDARD CONTRACTING ISSUES

1. **Most formally drafted contracts will deal with these issues.**
2. **Assignment**
   a. **Can the outsourcer assign the contract (have another outsourcer take over the contract?)**
      A. *If you sell the product to another company, or contract with another company to finish its development, is the outsourcer obliged to work with that company?*
3. **Integration Clause**
   a. **This clause says that the written agreement is the entire agreement and that no other understandings exist outside of this agreement.**
      A. *Courts enforce these clauses. If the outsourcer's salesperson made you a bunch of promises, but didn't put them in the contract, they aren't going to be enforceable against the outsourcer.*
      B. *It's not good enough to get a promise in writing. You have to get the promise in writing in a way that makes it clear that it is part of the final contract.*
4. **Choice of Law and Forum**
   a. **Governing Law**
      A. *Should the contract contain a clause that specifies which state's or country's law will govern this contract and the performance of the services*
   b. **Choice of Forum**

A. *Should the contract contain a clause that specifies which state's or country's law will govern this contract and the performance of the services? Why would you ever agree to a clause that sets the forum far from your own state?*
B. *Is the forum selection permissive (you agree that you CAN be sued in Washington) or exclusive (you agree that you can ONLY be sued in Washington)?*

5. **Relationship of the parties**
   a. **The parties are independent contractors**
      A. *No agency relationship.*
      B. *Not a joint venture or a partnership.*
      C. *Neither party can bind the other to any agreement.*
      D. *No tax withholding.*
6. **International legal issues.**
   a. **Barriers to using an out-of-country outsourcer**
      A. *Is there any problem with trans-border data flow? (There might be, for example, if you are sending customer data out of Europe.)*
      B. *Is there a problem with sending software across the border (e.g. encryption routines, licenses of 3rd party software, etc.) (Are there tools that you used to test the English version in the US that you can't let your outsourcer use to test the French version in France?)*
7. **Severability**
   a. **If part of the contract is determined to be unenforceable, the rest of the contract is to be enforced as if these clauses had never been in it.**
8. **Notice**
   a. **If you or the outsourcer must send legal documents to each other, does the contract specify the correct person to send the notices to?**

G. USEFUL REFERENCES

1. **American Law Institute, *Restatement (Second) of Contracts*, *www.ali.org*.**
2. **Bach, James (1997a) "Things to think about before outsourcing", downloaded from www.stlabs.com/outsource.htm, 3/25/97.**
3. **Bach, James (1997b) "Attributes of software testability", downloaded from www.stlabs.com/tptest.htm, 5/4/97.**
4. **Bach, James (1997c) *Pros and Cons of Outsourcing Software Testing,* ST labs Lecture Notes sent to me as a personal communication, March 1997.**
5. **Bach, James (1997d) *Outsource Vendor Analysis,* ST labs Lecture Notes sent to me as a personal communication, March 1997.**
6. **Classen, H.Ward, "Information Systems outsourcing agreement," in *Corporate Counsel's Guide to Software Transactions,* Business Laws, Inc.**
7. **Fisher, R. & Ury, W. (1991, 2nd Edition) *Getting to Yes: Negotiating Agreement Without Giving In*. Penguin Books**
8. ***FNS Mortgage Service Corp. v. Pacific General Group, Inc. and International Assoc. of Plumbing and Mechanical Officials* (1994), *California Appellate Reports*, 4th Series, vol. 24, p. 1564. (District of Delaware).**
9. **Freund, J.C. (1992) *Smart Negotiating: How to Make Good Deals in the Real World.* Simon & Schuster.**
10. **Halvey, John K., "Outsourcing contracts," in Kutton, L.J., *Computer Software: Protection, Liability, Law, Forms,* Clark Boardman Callaghan.**

11. **Hancock, William A., "The preparation of a statement of work," in *Data Processing Agreements 2nd*, Business Laws, Inc.**
12. **Hancock, William A., "Service Bureau Agreements," in *Data Processing Agreements 2nd*, Business Laws, Inc.**
13. **W. Hancock (Ed.) *Corporate Counsel's Guide to Software Transactions*, Second Edition, Business Laws, Inc.,**
14. ***Hanberry v. Hearst Corp.* (1969) , *California Reporter,* vol. 81, p. 519. (California Court of Appeal.)**
15. **Hancock, William A. (Ed.), *Corporate Counsel's Guide to Outsourcing,* Business Laws, Inc.**
16. ***Hempstead v. General Fire Extinguisher Corporation* (1967) *Federal Supplement,* vol. 269, p. 109.**
17. **Hiles, Andrew (1993), *Service Level Agreements: Managing Cost and Quality in Service Relationships,* Chapman & Hall.**
18. **Gelperin, David, "A testability maturity model," *Proceedings of the Software Testing Analysis & Review conference,* May, 1996, Orlando.**
19. **Kaner, Cem. (1996a) "Computer Malpractice" *Software QA*, Volume 3, #4, p. 23, available at www.kaner.com/malprac.htm.**
20. **Kaner, Cem (1996b) "Contracts for testing services: Warranty and indemnification," Software QA, Volume 3, #5, p. 20. available at *www.kaner.com/contrac1.htm***
21. **Kaner, Cem (1997a)"Contracts for testing services", *Software QA,* Volume 4, #2, available at www.kaner.com/contrac2.htm.**
22. **Kaner, Cem (1997b) *An Outline for Software Test Outsourcing,* Draft 0.87, www.kaner.com/outsourc.htm.**
23. **Kaner, C. (1997c) "The impossibility of complete testing", *Software QA*, volume 4, #4, p. 28.**
24. **Kaner, C. (October 8, 1997d) "Legal Issues Related to Software Quality", presented to the *Annual Meeting of the American Society for Quality, Software Division,* Montgomery, AL.**
25. **Kaner, Cem (1998) "Contracts for Y2K services: Traps for the software testing consultant," Software QA, Vol. 5, #2, April/May, p. 24.**
26. **Kaner, C. (1999) "Yes, but what are we measuring?", Invited address, Pacific Northwest Software Quality Conference, October 12, 1999; Presented as the focusing discussion paper at the 8th meeting the Los Altos Workshop on Software Testing, Cupertino, CA November 4-5, 1999.**
27. **Kaner, Cem, & Bach, James (1995) "Getting the most from outsourced testing" *The STL Report,* Nov./Dec..**
28. **Kubey, C. (1991) *You Don't Always Need a Lawyer*. Consumer Reports Books.**
29. ***MAI Systems Corp. v. Peak Computer, Inc*. (1993) *Federal Reporter*, Second Series, Vol. 991, p. 511, United States Court of Appeals for the Ninth Circuit.**
30. **McConnell, Steve & Tripp, Leonard (1999), "Professional Software Engineering: Fact or Fiction?" *IEEE Software,* November/December, 13, 16.**
31. **Milstein, Julian S. & Katz, Michael S., "Sample outsourcing agreement," in *Corporate Counsel's Guide to Software Transactions,* Business Laws, Inc.**

32. Moskowitz, H. & Wallace, R. (March 7, 1996) "Loser Pays. A Deterrent to Frivolous Claims." *The New York Law Journal*, p. 2.

33. Mylott, Thomas R. III (1995) Computer Outsourcing: Managing the Transfer of Information Systems. Prentice-Hall.

34. Nuara, L.T., H.P. Benard, & D.K. Rydberg (1998), "Year 2000: Problem or Opportunity?" in Kerr (1998).

35. Ocampo, Ray. L., Jr., Curtis, Shelley. S., & Moss, John. J. (1996) *Negotiating and Drafting Software Consulting Agreements,* Glasser Legal Works.

36. Raysman, Richard, & Brown, Peter, *Computer Law: Drafting and Negotiating Forms and Agreements,* Law Journal Seminars Press.

37. Reid, Warren S., "Outsourcing: The 20 Steps to Success," reprinted in Hancock, William A. (Ed.), *Corporate Counsel's Guide to Outsourcing,* Business Laws, Inc.

38. XXCAL Inc., *Why XXCAL.* Document downloaded from XXCAL's web site, www.xxcal.com/why.htm, on March 25, 1997.

Note: Several of the legal treatises above are undated because they are regularly supplemented (updated). These citations are appropriate to the 1999 supplements of the books, but the articles appeared substantially earlier.