

Legal & Practical Issues When Outsourcing Software Testing

Cem Kaner, J.D. Ph.D.

STAR 1997

Why is Outsourcing so Complicated?

- All sides often lack contracting experience (except the lawyers, who lack software experience).
- Developer/publisher engages in wishful thinking about schedules, capability & cost.
- Outsourcer promises too much.
- Outsourcing carries communication costs, forces (extra) documentation.
- Performance standards are unknown, unused, or not costed out in a rational way.
- Poor contingency planning.
- Conflict resolution process is weak/undefined.
- ***Developer/publisher fails to manage the work.***

What do I mean by outsourcing?

- Delegation of work to some organization out of your immediate control
 - might be an outside firm or another organization in-house
 - might be your entire test organization (all projects) or just a single project or just a single task on a single project
 - might be done with the goal of removing work from your company or with the goal of developing expertise inside your company
 - might be done to satisfy your own requirements or to satisfy a contractual obligation of independent testing.

Should we be outsourcing testing?

- General recommendations are to avoid outsourcing anything that is:
 - central to your business
 - dependent on a fundamental understanding of your business
- In discussing outsourcing of IT functions, authors suggest keeping your analysts, because they have domain knowledge that is specific to your business.

(Hey, does that imply we should NEVER outsource testing? That we should be very, very careful about when we do it?)

What should your outsourcing contract include?

- You want a formal agreement, even if you are delegating to another part of your company.
- You can't plan for every contingency. You want
 - a broad agreement on scope, objectives, and schedule
 - clear agreements about money and authority
 - a solid dispute resolution procedure.
 - If you know more details than this, include them. At some point, though, you have to shake hands and start working.
- Lawyers often draft contracts from checklists. There are lots of IT and Support outsourcing checklists but none (that I've found) for testing. Here is an overview of my current draft. For more detail, see my paper in Software QA Magazine.

FUNDAMENTALS

1. Why do you want to outsource?

- you're running too many projects & need to focus
- the project is well organized and easy to outsource
- you don't have enough staff and can't recruit
- you can't do the job fast enough
- you're missing needed expertise
- you're missing needed equipment or software (config testing or testing tools)
- you believe the outsourcer will be more efficient
- you want the outsourcer to train your staff
- you need an independent advisor (severity, completeness, etc.)
- contractual obligation to outsource (independent testing)
- you want certification
- some yahoo in senior management "knows" that outsourcers do it faster, cheaper, better. (*"Let's quit worrying about testing."*)

2. *What is the scope of the work?*

- Entire testing department (*We give up testing.*)
- Entire platform (*We give up testing Macs.*)
- Entire product line (*We give up testing word processors.*)
- Entire product (*We give up testing BuggyWord 5.0.*)
- Maintenance release (*We give up BuggyWord 5.01.*)
- Entire language (*We give up BuggyWord in French.*)
- Class of issue (*We give up testing printer compatibility. We give up testing against IRS regulations.*)
- Buy skill (*Give us a test plan or an automated test suite and show us how to use it. Show us how this tool works.*)
- Additional testing (*We want audit / usability test / sanity check.*)
- Required independence (*We have to use a *&^?! independent test lab.*)

3. *What is your schedule?*

- *You want to ship it next Tuesday?*
- Are you being told to use an outsourcer to make it “possible” to achieve a fantasy schedule that the rest of the company won’t make anyway? Watch how this distorts your cost structure:
 - rush charges to ramp up
 - additional charges for going beyond the planned # of cycles of testing
 - penalties for tying up key staff past planned project end (or more ramp up cost)
 - all the inefficiencies of no-time-to-plan-just-do-it testing.
- If the schedule is unreal, maybe use the outsourcer as the independent investigator who explains that to your management, and then pull the *testing* back in-house.

4. *Who has final authority on priorities?*

- How independent is the outsourcer?
 - Is independence required by contract?
 - If not, you have pragmatic tradeoffs to make between the (hopefully) expert testing judgment of the outsourcer and the (hopefully) informed judgment of your staff as to what features and issues are important to your customers.
- How loose are your in-house cannons?
 - Is the product safer in the hands of your outsourcer? If so, can you work a deal in which the outsourcer understands that it is to be “hard nosed”?

5. How do you want to pay for this?

- There are choices
 - hourly
 - flat-fee
 - flat-fee plus time-and-materials
 - flat-fee with incentives and disincentives
 - piece-work
 - bugs?
 - function points or KLOC?
 - configurations?
 - what else?
- Each type of fee structure is open to abuse
- Hourly contracts give you more control than flat-fee

6. *How will you resolve disputes?*

- Why is this so important to resolve upfront?
 - a good process lets you work with less tightly defined deals.
 - typical processes (lawsuits, arbitrations, putting up with it) are unsatisfactory.
 - *Resolving disputes can be done in a friendly way, if you plan for it.*
- Ideas on resolution, *mid-project* (before it's too late)
 - meet and confer
 - escalating discussions through management
 - mid-project mediation
 - independent mini-arbitration
 - independent fact-finder
 - *Agree up-front about these third party helpers*

DETAILED OBJECTIVES

7. What specific jobs do you want done?

- You can split the work any way you want. Make a list of tasks. (See “Negotiating Testing Resources” at www.kaner.com.) Which ones do you want to farm out?

8. What do you want to learn from this?

- If you want training, you have to plan for it.
 - Testing or test planning techniques? Standards? Tools? Efficiency-improving tricks and forms?

9. Can this outsourcer meet your needs?

- Staff (*By the way, will you be assigned the people you meet?*)
- Lab facilities
- Tools and experience with these tools
- Experience with your type of application

COMMUNICATION & CONTROL

10. How will you supervise the effort?

- Your company **MUST** supervise the effort.
 - replicate all bugs and look for more serious consequences.
 - audit (don't just review) the test plan. (How will you find holes.)
 - look for communications problems
 - monitor progress
- Your company needs a **LIASON** to supervise / talk / settle problems with the outsourcer.
- Your company's **LIASON** needs sufficient time to do a competent job.

11. What status reporting do you expect?

- Of course you want regular status reports.
- (How much do you want to pay for them? Detail = money.)

12. What is the bug reporting/resolution process?

- Whose bug tracking system, yours or theirs? Why?
- What is the bug lifecycle from report to closure? Does it include as many checks / safeguards as in-house bugs?

DELIVERIES

13. What are you supposed to provide?

- software? hardware? test plan? test cases or data? tools? people? (*And will you get any of this back?*)

14. What is the outsourcer to provide?

- software? hardware? test plan? test cases or data? tools? people? (Make a list of everything that you are to receive.)

15. What stuff is coming from 3rd parties?

- These are the things that aren't under your or the outsourcer's direct control. List them. Who will manage these?

RISK MANAGEMENT

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

- Do they know your market?
- Do they have an equipment lab?
- How junior are their staff?
- Who will do your glass box testing?
- Can they measure coverage for you?
- How well do they understand your application domain?
- How well do they understand your environment (O/S, etc.)?
- ***Break out all of the tasks and work with the outsourcer to appraise where they are strong and where they are weak.***

17. How will you deal with “unexpected” contingencies?

- What if the outsourcer completely screws up?
- What about the unexpected (if inevitable) last-minute shuffling of the feature set?
- What if the software is untestable?
 - For how long?
 - What do the outsourcer’s staff do in the interim?
- What about staff turnover?
 - *(How much are you willing to spend to manage this risk in advance? Maybe you will choose to leave this an open risk.)*

18. What performance standards are there for quality of work or product?

- Standardless review and signoff?
- Standardless independent audit?
- Conformity with industry or internal standards
 - IEEE 829 for test plan structure/content
- Specific standards
 - not more than X% irreproducible
 - turnaround on queries within X days
 - 5 bugs per day? (not recommended)
 - test cases coded per day (but how powerful?)
- What about a bonus system based on what bugs were first discovered in the field or were recognized in the field as more serious than implied in the bug report?

19. What liability is there for quality of work or product?

- Can you sue the outsourcer for bad work?
- Can your customer sue the outsourcer for a bad product?
 - The problem of certification

20. What else is your lawyer demanding?

Some common issues raised in contracts can kill deals, even if you don't think they're important.

- Nondisclosure and security
- Noncompetition
- Indemnity
- Liability insurance
- Legality of use of 3rd party software
- *Who owns the results?*