Print Article Close Window

From: www.cio.com

Are Purchasing Practices Killing Your Software Projects?

– David Taber, CIO

January 30, 2014

The larger the organization, the more comprehensive and sophisticated the purchasing department. By setting up highly optimized processes, purchasing saves your company time and effort for 80 percent of purchases.

But there's a big issue for IT: There are no companies where 80 percent of overall purchase volume consists of custom software or system integration. By definition, purchasing processes, methodologies and standards are suboptimal for software projects that use consultants or integrators.

How sub-optimal, you ask? Take a look at this top-10 list of contractual issues I've seen in the past year. Not one is made up. How much will this list impact your project? Let's leave that unhappy news for the end.

Top 10 Stupid Software Contract Tricks

10: Requiring the consultant to transfer ownership of the work product. This makes perfect sense to the purchaser, but abiding by it really means setting up a clean-room development environment (to avoid intellectual property contamination) and re-inventing the wheel for each and every feature your project requires. That means you get a unique piece of code, with unique bugs.

[Analysis 7 Top Wishes of IT Project Managers]
[More: 3 Ways to Be More Agile With Software Shipping Decisions]

If, instead, the consultant retains ownership of the IP, your project can take advantage of all the labor of its previous projects and all the reliability from previous implementations. Let the consultant actually own the IP and you get the full source code, as well as a perpetual, irrevocable, fully-paid license to use the executable.

9: Establishing legal venue at the client's headquarters even though you have operations in the vendor' HQ jurisdiction. Everybody wants to avoid any hint at legal action. That's just bad news all the way around. The whole point of a contract, though, is to set the rules of engagement should an action be required.

Of course, you *want* to have any legal action filed near your headquarters' lawyers. We all live to improve *their* lives. However, your consultant probably isn't headquartered in the same state as you and may not have any formal operations there. The only real power you have to compel them to appear in court is your ongoing business — but if things are going that far south, that incentive is pretty weak.

Unless you're lucky enough to fit "sister state" reciprocity criteria, your local court's judgment would be ineffective anyway. Instead, you'd actually have more power by filing any action in the state where the

consultant is headquartered.

8: Requiring consultants to indemnify you without limitation, but not indemnifying them. This also makes perfect sense to the purchaser. Any vendor action that causes a third party to sue should sue the vendor. If you're not offering the reverse coverage, though, for actions that your company might take, then this is just a dickish power play. The net result: The consultant has to take out a lot more insurance, at a cost that's higher than your own insurance, driving up the rates charged for your project.

7: Requiring the consultant to finance you. This is a favorite of finance departments, but it creates a perverse cost function. As a large company, you have much lower capital costs than a consultancy. This requirement burns up a consultant's credit line. Plus, if you have payment terms of 120 days or more — further compounded by invoice-approval cycles involving people who know nothing about the project or software in general — you also cause cash-flow issues for the consultancy. In extreme situations, this results in staffing problems for your project.

6: Making non-poaching agreements a one-way street. Any consultancy is willing to sign up for not poaching your staff. If you aren't willing to make the same commitment, you give consultancy an incentive to put crummier staff on your project.

5: Not allowing explicit project management or client-meeting tasks in the schedule of work. This is non-thinking at its best. Any project must be managed, and you'll be calling meetings and participating in system walk-throughs. To not allow these as explicit tasks on the SOW means they'll be buried in other items, leading to misleading measurements. Remember the old project management saying: "You can't fix what you can't measure."

4: Heads-I-win, tales-you-lose pricing. I go on endlessly about the perils of fixed-price projects; namely, how they can poison the agile methodology that's the core of lowering project costs. Asking a consultant to absorb the risk of fixed price can mean doubling the bid.

[Related: Has Agile Software Development Gone Mainstream?]

Some clients take it a step further with "hourly rates, with not-to-exceed" clauses. This makes perfect sense to every purchasing manager in the world, but it contaminates your project with sloppy thinking, gamesmanship and an adversarial relationship. Agile requires trust. If you aren't willing to start there, go back to waterfall.

3: Requiring the consultant to cover you for patent infringement on patents that don't even exist yet. Patent trolls are a grim reality, yes, but it's a bit much to require a consultant to cover infringement costs and all lawyer fees for patents that you haven't issued yet (pending ones are secrets) or can't read (because they're in a foreign language). Most firms can't buy insurance coverage for this at all, so if a huge infringement judgment ever happens, all you'll do is bankrupt your supplier. That'll help.

2: Mandating clumsy, overly detailed project reporting that the project manager doesn't have time to understand. Measuring the wrong things doesn't help. It can actually cloud visibility. There's simply no substitute for a strong, trusted project manager with enough time and energy to keep the project schedule and budget on track.

1: Relying on a requirements tome, not a strong project manager. The RFP process works for hardware systems. It means almost nothing to success of custom software — and for cloud projects, I'd go as far as to say that they contribute to project failure.

[Related: Agile Project Management Lessons Learned From Texas Hold'em]
[More: How to Deal With Software Development Schedule Pressure]

Unless you're willing to invest an enormous amount of money on an exhaustive requirements spec, the RFP documents will be full of bad assumptions, misunderstood requirements and overblown details that all conspire to make your project a mess. With larger projects, many of the pages are obsolete or invalid by the time you implement them.

A small, flexible specification of goals with a dedicated, knowledgeable and strong project manager is a much better approach. Paper can't protect you from scope creep, misunderstandings and outdated goals, all of which contribute to project overruns.

Small Software Projects Don't Need Big, Onerous Contracts

The larger your company, the more likely you are to use these contractual tricks. In procuring large, long-running projects from large consultancies, such contracting issues can be easily amortized and hidden. Other than the high price tag, you may not even notice the effects on those projects.

[Tips: How to Change Software Testing for New Cloud Configurations]

Fine — but smaller projects, typical in IT's new cloud application paradigm, don't fare so well.

- You'll drive up project costs. The "minimum feasible project size" is probably $20,000. (Larger consultancies may not answer the phone for less than $500,000, but boutique shops can make a profit on much smaller projects &mdash if you'll only let them.)
- You'll lengthen the project schedule, as all the tech must be built from scratch, not from than recycling code or other IP.
- You're less likely to get bids from specialist shops where the proportion of sharp-shooters is higher. The average staff quality on your project will be lower.
- The inflexibility mandated in your contract means the project can't be as tight a "fit" to evolving requirements.

If you're trying to build software for today, and even tomorrow, you can't expect to succeed if you're still using the contracts of yesterday.

David Taber is the author of the Prentice Hall book, Salesforce.com Secrets of Success, now in its second edition, and is the CEO of SalesLogistix, a certified Salesforce.com consultancy focused on business process improvement through use of CRM systems. SalesLogistix clients are in North America, Europe, Israel and India. Taber has more than 25 years of experience in high tech, including 10 years at the VP level or above.

Follow everything from CIO.com on Twitter @CIOonline, Facebook, Google + and LinkedIn.