

# ;login:

THE MAGAZINE OF USENIX & SAGE

October 2000 • volume 25 • number 6



inside:

SYSADMIN  
Vendor Contract Services



## USENIX & SAGE

The Advanced Computing Systems Association &  
The System Administrators Guild

# vendor contract services

## A Primer

I'd like to talk about the logistics of coordinating large-scale service build-outs when you have several independent vendor teams and subcontractors working on a project. More and more of the "architecture" work I do is devolving into 30% architecture, 50% project management, and 20% keeping the finger-pointing and name-calling between vendor teams at a minimum. I know that I'm not the only one in this situation, and I'd like to share some of the things I have learned along the way.

I have been both an employee and a contractor for clients, and a subcontractor for a vendor team. I've managed projects involving vendor teams, managed vendor teams for a client, and managed vendor teams for the vendor. In other words, I have toured the sausage factory from most conceivable viewpoints, including that of the sausage. I won't say that I will never eat another sausage, but I will say that "managed eating" is a Very Good Thing when it comes to sausage.

I will also say that while there are few "right answers," some of the more correct answers from a client side run headlong into best practices from a vendor side, and vice versa. You may be frustrated at the lack of a clear recommendation as to the best course of action, as am I. The best that one can do in this situation is to illuminate some of the issues and tradeoffs, and allow each person to make the choices that seem best for their situation.

## Vendor 101

### THE STATEMENT OF WORK

First, understand that vendor services groups, unlike many individual contractors, generally have a well-defined process for engaging in work. They may only deal with firms as part of a product sales process, or they may deal with firms that have purchased products in the past. They tend to require a signed and approved Statement of Work (SOW) before letting the client talk to anyone but a sales person. The SOW is basically a contract defining the client and the vendor obligations. A good SOW will protect you, the client, with an appropriate level of detail about deliverables and about the process whereby deliverables are approved. Since the SOW must be approved before the work begins, and contains the financial information controlling the work, getting a good SOW in reasonable time is something of an exercise in compromise. Specifics will be detailed in the next section of this article.

Be aware that your vendor may look like a large, homogenous, well-managed corporation, but in fact may be a tangle of different subcompanies that have been acquired, autonomous departments, software/hardware organizations with completely different tech support and vendor services groups, etc.

To make things more interesting, even within one company or group, a vendor may have two distinctly different vendor services departments. One will be a sales department that books an SOW, and the other will be a deployment department that is basically stuck trying to deliver what the sales folks have promised. If your vendor sells hardware as well as software, it is extremely likely that there will be *at least* two different vendor service groups, hardware and software, which may also contain this dichotomy within them. Just because you have signed on an SOW does not mean that the vendor

by **Strata Rose Chalup**

Founder, VirtualNet; Consulting Director of Network Operations, KnowNow Inc. Strata Rose Chalup got her first sysadmin job 1983 and never looked back.

<strata@virtual.net>



actually has personnel available to assign to your work. It is generally a true sales process, meaning that the goal is to get you to sign first and then to deliver afterwards.

Most vendor services groups I have seen seem to be constantly shifting between a one-structure and a two-structure arrangement. The truth of the matter is that vendor services are hard to deliver, since you are dependent on the usual intricacies and delays of the sales process plus the real-world constraints of personnel availability. There is no “perfect fit.”

### THE VENDOR TEAM

Many firms are in a constant battle to retain personnel and to protect them from burnout. Being a vendor services team member generally means experiencing most of the angst of consulting (variable schedule, constant emergencies, frequent travel) without the benefits of consulting (high rates, vacations between contracts, freedom to turn down work). It doesn’t take a rocket scientist to figure out that a move into actual consulting or into another department or company can result in higher compensation, reduced job pressure, or both.

Your vendor team will probably be composed of vendor employees and of contractors. Either the employees or the contractors could be people who have worked extensively with the products, or people who are new to the vendor and whose resumes indicated product familiarity. If the team member is new to the vendor, he or she may not have the crucial engineering and support contacts needed to resolve problems with your deployment. As a contractor myself, I do not wish to tar vendor services contractors with any sort of broad brush. Most of the ones I have worked with have been talented professionals who have conducted themselves admirably at client sites. In practice, however, it is more likely that the vendor services employee has established relationships with key engineers and high-tier support personnel. He or she can bypass cumbersome tech-support protocols with private email to internal developers, search the employee bug database, etc. It is a very good idea to make sure that for each major vendor product in your deployment plan there is one person, employee or contractor, who has a strong history of working tightly with the vendor on that particular product.

As I have often discovered, the right hand rarely shares resources with the left. A truly dynamite vendor team member who has access into a vendor’s VPN product may have little or no access to the same vendor’s firewall product group. Familiarity with the engineering group responsible for the messaging product in no way guarantees knowledge of the engineering group responsible for the Web-server product, etc. This is of particular importance when dealing with a multi-product deployment from a large vendor. Some of the products were probably developed in-house, some may have been acquired as intellectual property, and some may have been acquired along with the company that developed them. You have no visibility into the internal support structure of the vendor, so you

must ensure that the expertise will be available to you via your vendor team regardless of the strange politics that may hold sway behind the scenes.

Role	Hourly Rate	Duration
Project Manager	\$250	12-14 weeks
Messaging Architect	\$225	2-4 weeks
Messaging Developer	\$175	4-8 weeks
Scripting Developer	\$175	2-4 weeks
Firewall Architect	\$200	1 - 2 weeks

*Table 1: Sample project assignments*

## DEPLOYMENT AND FINANCIALS

In the vast majority of cases, deployment of the vendor team will be assigned and specified by role and duration. Where this is done, rates will be assigned via roles, rather than individuals. An example is given in Table 1.

Given personnel changes and project constraints, it is very unusual for a vendor services organization to list specific individuals as committed to a project in the SOW. If your firm has experience with any specific individuals and wants to require that they be part of the team, it is necessary to capture that quite explicitly in the SOW or in an addendum that is indicated as binding by the SOW. Issues surrounding role-based resources vs. individuals are discussed more thoroughly in the “Resources” section below. The Statement of Work will generally require a financial commitment for the maximum estimated time listed for a given role. Be certain that your vendor’s SOW indicates that only the actual time worked will be billed, rather than the entire allocated duration of the role.

An overhead figure will be set in the SOW for incidental vendor expenses, including reimbursement for travel expenses for personnel who are brought in from out of town for the duration of their role in the project. Typical numbers are 12% to 15% for overhead. Note that if your firm is located in an extremely expensive urban area such as San Francisco or New York City, this figure may be higher. You may be familiar with “overhead” from a university or research environment, where it is deducted by your parent organization from the total monies allocated. The type of overhead expense mentioned here is only billed against to provide direct reimbursement for expenses incurred by the vendor on the project.

That said, it is up to your firm to track these expenditures closely and ensure that they do not exceed the figures budgeted in the SOW. The vendor organization does not typically track expenses vs. overhead allotment, relying on the client to protest if the expenses exceed the allotment. It is often helpful to request a copy of the vendor’s travel and expense policy for employees and subcontractors. The vendors generally review individual team-member expense reports against the criteria in the policy.

Be aware that it is very common for vendors to pull in employees and subcontractors from other regions in order to fill out a team. Most projects have a rate structure such that the vendor can profitably employ subcontractors from other regions as long as there is an overhead for travel expenses paid by the client. Do not assume that the overhead will go unused if the vendor is local to your area or has a regional office nearby.

## Client Requirements

### KNOW YOUR OWN CONSTRAINTS

First, ask yourself the Very Hard Questions: Can anyone really get this project delivered in that timeframe? Do you have any idea of the complexity of what you’re doing? Have you read *The Mythical Man-Month*?

We can talk about that in another article, but for now we’ll stick to the rack-space rental, bringing in your WAN lines, setting site policy, hiring your ops staff, setting up your trouble-ticket system, setting up your tier-1 support including 800-number help desk, etc. Okay, maybe your project is different: small, manageable. Probably not, since companies never seem to call vendor contract services for those kind of projects – they just overload their sysadmins. Or worse, they may not even have enough of a realistic schedule to get beyond the “I don’t care what it takes, just do it” mentality. Marketing has been planning this for a year, engineering has been writing the custom app, and now, six months before launch, the exec-level project plan says “hire a director of oper-

First, ask yourself the Very Hard Questions: Can anyone really get this project delivered in that timeframe?

You should have a distinct plan for what you wish the vendor team to accomplish.

ations and commence buildout of site.” You have my deepest sympathies. Exec staff always seems to believe that a big enough budget can change the law of physics.

Are you doing this to spare your folks the ramp-up time on getting things working, or are you secretly hoping to offload a whole bunch of stuff onto the vendor team and use them as generic sysadmin help? Are you really prepared to give them a *working* infrastructure or are you building everything in parallel and wasting a lot of their time (which you will pay for!) while you fumble with your network plan?

#### DEFINE YOUR GOALS

You should have a distinct plan for what you wish the vendor team to accomplish.

Note that if what you are trying to do is very complicated, you may find limitations in the products or their interactions of which you were formerly unaware. For this reason, I recommend purchasing the minimum number of licenses or copies of the software required to establish a working test setup. If you are integrating a number of different vendor products, you will undoubtedly find limitations in their interaction that will require additional engineering work on your part. You may even find limitations that call the validity of your entire production model into question.

A typical e-commerce project may be attempting to integrate new accounts, provisioning existing/new accounts, billing, Web forms, messaging, security via certificates, advertising, and a store or sales engine. Companies or products involved might include PeopleSoft, Oracle, Oracle Financials, custom scripting, LDAP servers, LDAP gateways between PeopleSoft or Oracle and the LDAP servers, messaging servers, mailbox provisioning with “welcome to this” messages, certificate initializing, setting uniform passwords across services, etc. There are so many places for things to go wrong that it boggles the mind.

If you have an unusually strong requirements document and someone to drive the project to the level of detail necessary, you have an excellent chance of finding out what works and doesn’t work during the design and architecture stage. If not, you will have to find out in reactive mode while trying to meet your deadlines, and go through annoying and probably expensive change control with your original SOWs and project plans while building the live system.

## Constructing the Statement of Work

#### PROJECT PLAN VS. SOW

To build an SOW, you should have real deployment plans, with clearcut deliverables for your team as well as for the vendor team. The usual catch-22 is that you generally need the assistance of the vendor team to make those deployment plans. It would be wonderful to have an actual project plan delivered as part of the SOW, especially for these big \$250K-to-\$500K projects that involve a whole vendor services team. This seems eminently sensible from the client viewpoint. Your vendor manager will balk at this, since developing project plans is usually a source of revenue and part of the service provided. If you ask for a project plan as part of the SOW, you are asking him or her to commit an expense for “business development” to create that project plan, i.e. to adversely affect his or her bottom-line numbers.

You could attempt to insist that the project plan be listed as part of the SOW, but delivered *before* the SOW is approved. You will not get this kind of concession from an experienced manager, as you are asking him or her to commit resources before any legal guarantee of payment is in place. This is reasonable, as lack of an approved SOW also

exposes your firm to the risk that personnel will be needed elsewhere and reassigned, leaving you in the lurch.

In theory, it would be better to insist on separate SOWs for the deployment plan and the “real work,” so that the mutually agreed-upon deployment plan can serve as the basis for estimations. In practice, this means committing to an architecture SOW separately from a deployment SOW.

Be aware that without a detailed project plan, the vendor account manager will be unable to get accurate estimates from the vendor architects and potential team members. This could lead to a highly inflated fixed price, or the imposition of time and materials with an unreasonable cutoff point.

#### **MULTI-STAGE VS. MONOLITHIC**

Clearly we are back to our catch-22. I still believe that the best way to proceed is to arrange a fixed hourly allotment for preparation and review of a project plan, including identification of resources required by the vendor team. This “mini-SOW” will get things off on the right foot by getting you a firm set of plans and deliverables. It will almost certainly need to include architectural design and review unless you are merely hiring vendor implementation for a predefined architecture. Most contract services groups will pressure you to do one big SOW that includes architecture and deployment, but I caution you against this. While the vendor may present some compelling arguments, most of them boil down to “I really want to book this.” Be aware that the manager who books your SOW receives a very substantial commission bonus in many, if not most, vendor contract services organizations. In a split sales/deployment contract services organization, the manager may even be subject to quotas or deadlines similar to those in the main product-sales group.

Be aware that approval of an SOW on the vendor side may be a one-to-three-week process, as it has elements of a binding contract and must be run past the vendor’s legal department. A “dead zone” of one to three weeks between the project-plan phase and the deployment phase may affect the availability of the vendor team, since there are usually more projects than personnel to implement them. Creating the larger SOW in parallel with the mini-SOW, incorporating its deployment plans as they come in, is the ideal approach. Tip: If you and your vendor manager pre-approve the wording of the SOW shell first, then add specific deliverables, you can shave days or weeks off the approval process when you have your final large SOW draft ready.

You can usually get a good feel for the availability of team members from the account manager, and should plan multiple vs. multi-stage SOW from there. If a vendor manager claims that he or she cannot guarantee that you will get the resources that you need unless you commit now and schedule their time out N weeks, I would think seriously about another vendor. What that manager is really telling you is that they have a very disorganized or loosely organized department that is long on managers and short on technical talent. In those environments, managers with paying work get to reserve architect and senior-class people, usually through some project-tracking system.

If your choice of vendor is dictated by the product(s) you have purchased, you are between a classic rock and a hard place, and you need to decide for yourself how much is bluff and how much is real. Attempting to close quickly on a “mini-SOW” is probably the right choice, since your vendor manager is also looking at lost revenue if he or she can’t pull a team together.

If a vendor manager claims that he or she cannot guarantee that you will get the resources that you need unless you commit now and schedule their time out N weeks, I would think seriously about another vendor.

## RESOURCES

Make sure you get commitments of specific personnel with specific experience, not “or equivalent.” All vendor services organizations are subject to pressures that encourage them to pull architects and senior folks off and put you in a sustaining role before you are really ready. You *want* those senior-level people around when you start testing, for instance, since they can debug problems much faster than the junior folks. And when the SOW says “equivalent,” it means you’ll be paying the same dollar for those junior folks as you would have if you’d stuck to your guns and kept your original team. Watch out for this one, it is a big one and quite common. Just smile sweetly and say that if so-and-so is an “equivalent resource” then he or she can go off to the East Coast to do the work that your original architect is supposedly called away to perform. After all, they’re “equivalent” aren’t they?

There are some legitimate situations in which you may need to okay resource substitution for a period of time. Perhaps your senior implementor came to your project from a six-month gig at BigCompany and they have run into a problem. You can understand that he or she could solve the problem faster than an equivalent, due to the prior familiarity with the site. You hope that if you run into problems down the road that you would be able to command the same priority on a few days of the person’s time. That’s fine, and is part of the inevitable give and take. It’s when the vendor wants to start a new project with your paid resource that you should be prepared to hunker down and not give in. If you do agree to a substitution, specify a maximum number of days, after which you must reapprove the substitution in order for the vendor to be meeting the contract terms.

Part of the SOW should include measurable, quantifiable milestones and checklists for determining when those goals are met, so neither you nor vendor team will say “but it isn’t done.” If those checklists are not available at the time of the SOW, you must make them an explicit deliverable with a signoff stage. You must also be willing to acknowledge any of your own mistakes, and pay for them if necessary. An example from real life: the network is stable for several weeks, then the client decides to rip out the core router and rewire the racks *after* the vendor team has conducted performance testing. Are you sure the new configuration is stable enough to let the old performance testing stand? What if the new load balancers you are introducing don’t really do a valid RSET on expired cached connections, and thus confuse the LDAP connection caching on the mail relays? Are you going to find out during your beta test phase that if no email passes through the system for five minutes or more, the mail relays get into a wonky state? A typical true, if ugly, story.

## DOCUMENTATION

One common “gotcha” is documentation. The SOW should specify not only documentation on any software installed by the team, but also documentation of config files. These should include annotations and/or explanations in or with the file, not just printouts of the file contents. It should also be explicitly stated that any scripts, programs, or utilities brought onsite by the vendor team to do their job should remain installed onsite *and* be documented by the vendor team, even if those tools are not official products of the vendor. I would advise that the wording indicate that tools explicitly published/provided by outside groups such as the Free Software Foundation are exempted from this requirement, as are tools published by individuals outside the vendor group. Any modifications to those outside tools done by vendor group employees or contractors, however, even if performed outside the duration or scope of your project, should be required to be documented.

## OUTSIDE TOOLS

To protect yourself, I also advise inserting a clause that indicates that any licensing or usage fees, copyright issues, or intellectual-property issues involving tools, scripts, or utilities brought onsite or requested/required by the vendor team are the responsibility of the vendor and that you will not be liable. If that nifty set of host file to LDIF scripts turn out to require an annual fee if used commercially, for instance, your vendor should not be able to say that those scripts are crucial to their completion of the SOW and that you'll have to pony up. If they say that *before* the SOW, that's fine, at that point you and they can negotiate for the extra time and effort needed to work around or duplicate that functionality. What you don't want is to find out halfway through the project that you're in violation of someone else's shareware licensing.

## CONTROLLING SCOPE

Note that your vendor services account manager is almost certainly subject to "variable compensation," i.e., a commission. He or she will be genuinely helpful, but also attempt to sell as large a contract as possible. Be realistic about what portions of the job can and will be done by your staff, and what portions should be implemented by the vendor team. If you assign too much to your team, you will be paying for the vendor team to twiddle their thumbs while your team struggles to complete dependencies so that the project can move forward.

One of the most helpful practices in this area is to encapsulate functional pieces of the project such that you minimize interdependencies but allow both teams to proceed in parallel. An example of this would be specifying that your team will be providing the machine infrastructure and firewall, whereas the vendor team will be installing and configuring their application servers on the machines using the assigned names and IP addresses, then cooperating with your team to assist in the final testing of the firewall.

It is vital to make certain that the SOW explicitly includes all stages of the product-deployment lifecycle, since the vendor team will be using the SOW to determine their duties. Many items are commonly forgotten or merely assumed by the client, and there is friction introduced when differing expectations collide. Do you have a test lab? Will the vendor team be installing the products on the test lab first, testing them, then installing them on production hardware and re-testing? Is the testing simple functionality testing, or is it performance testing as well? What are success characteristics for the installation, and are they functionality-based, performance-based, or both?

Recognize that most vendor services teams' first goal is to get the job done. Their very close, but still second place, goal is to make you happy. Question, question, question, and then question some more. Make sure your site practices and standards are fully captured in the SOW, so you don't end up with a vendor team insisting that it's your responsibility, for instance, to make Jumpstart scripts for their packages since the SOW only says "configure" and doesn't spell out "provide working Jumpstart configurations."

Review SOW architectures with the vendor team leaders in a wider sysadmin context. Many of the architecture-level vendor team leads I have met are actually fairly recent CS grads who are very sharp in their vendor's product but have little or no clue about sysadmin "best practices." They tend to produce vendor architectures that are excellent from a high transactions-per-second throughput-rate perspective but poor from a redundancy, maintainability, and disaster-recovery perspective. For example, little or no attention is generally paid to preventive-maintenance issues such as alert levels, log rotation, etc, leaving your team to reconfigure (and possibly break) the vendor packages after the vendor team has signed off on them.

What you don't want is to find out halfway through the project that you're in violation of someone else's shareware licensing.



## Working Together

### PROVIDE APPROPRIATE INFRASTRUCTURE

Vendor rates and time estimates do not generally include spending half a day here and there getting FTP to work between your colo hosts and your internal network, for instance. It's up to you to have those things in place. It's perfectly legitimate to ask the vendor team to provide a list of what facilities they need upfront, though, and to insist that the list be meaningful rather than vague. A typical response might be "normal system functions available," to which you might push back with a list of services and checkboxes as to which are truly required. Be aware that many vendor teams rely on Internet access to pull binaries and patches from their own sites, so be certain your list specifies "internal" vs. "external" access for services such as Telnet, ssh, FTP, and email.

A savvy vendor services team will come with a project manager, who will insist that any delays caused by your infrastructure be included quite explicitly in the project plan, and who will bring change control processes to your management if presented with a non-working environment. Be aware that multiple "little things" that seem inconsequential to you will add up to significant time spent wrestling with your environment. If you succeed in obfuscating the issue, all you are really doing is delaying the inevitable, and pushing the argument off to the sign-off portion of the project. If the vendor is doing time and materials, the higher cost will come out in the end. If there is a fixed-price contract in place, certain items like documentation and testing phases may end up being silently ignored or handwaved around because of the extra time taken previously. Yes, it is unprofessional. Yes, it happens all the time, especially once the servers are working and your management has lost interest in chasing the vendor for some missing docs.

### TEAM INTERACTIONS

It's a good idea to hold cross-vendor "global group" meetings, with your own people and the vendor team. Obviously if you have all of the vendor team members in your weekly staff meeting, you are paying a lot for them to sit around and eat donuts with your team. Structure regular attendance at your staff meeting for their project manager and technical lead.

Don't let your team drag the vendor team into kibitzing on things that aren't their SOW. Many of the folks on the vendor team will have relevant experience that could benefit your group, but you must be absolutely clear to both your team and the vendor team that they are hands-off! Be certain that the vendor team manager is very clear on your policy, as he or she may need to remind individual team members that they are not allowed even to "take just a quick look" at something outside the scope of the work they are doing.

Do plan for your team to spend some unstructured time, such as a team lunch or coffee break once or twice a month, with the full vendor team to get the benefit of their experiences on projects like yours. An hour of trading war stories not only builds rapport but gives your team an idea of who actually knows their stuff on the vendor team and who is just full of it – and vice versa! Some of your own team may not be as strong as you'd like, and a good relationship with vendor project managers or technical leads can often yield information about your own people as well.

Brief your team and your support organization about the vendor team's existence, their project, and its priority in the scheme of things. If you are the member of a networking group in a large company, working with a switching vendor on your e-commerce site,

it's likely that the folks at "support@mycompany" have never heard of the vendor team and will sit on their requests for a couple of days trying to figure out who they are and to which department they report.

#### UNOFFICIALLY MANAGING VENDOR TEAMS

You may or may not be your firm's client contact for the vendor team. You may be a technical lead who is working closely with them. Or you may be a manager to whom the vendor team reports as a whole, but individual vendor team members report to their own project manager. Regardless, there are things you can and should do to "manage" the vendor team and provide some guidance.

The first thing is simply "be nice to them." Yes, you've got better things to do than coddle other people's employees that you're hiring at \$150+/hr. Usually, so do *their* managers. Most professional-services engineers are overworked, underappreciated, and on their own with little management support. Their managers are usually out lining up gigs for them that demand weekend travel or 24x7 on-site presence. If you treat these folks more like your own team than strangers, often they will jump through hoops for you and that may make the critical difference for your project.

You can provide some extra support, give positive feedback whenever appropriate and be considerate in giving negative/constructive feedback, include them in some of the lunch runs and T-shirt giveaways, etc. They will usually take an interest in your project and begin to think of you as a friend and colleague more than just another client to survive. Miracles can and do happen, ranging from custom features finding their way into a release to client project staff being put on the "friends" list for an IPO. (That last is a true story from a recent project. I was on another vendor team, alas, and arrived on-project very late in the game. Oh well! Maybe next time!)

On the flip side, prepare to spend what seems like endless and needless times nudging vendor team members through fairly simple roadblocks. They are usually operating under stress, so often their tendency is to want to fingerpoint, quit for the day, and go back to the home office to catch up on paperwork, email, etc. It's not malicious, it's simply a response to things that seem to be out of their control, so they are going to exercise good time-management skills and do something else.

Make sure that someone from your team is always available to the vendor team so that he or she can catch these things in action and insist on trying to walk through the obstruction, fix the problem right then and there, make internal requests within your organization, and generally grease the wheels. You are probably on a tight schedule, and you'd rather have one of your team members call the Security group and arrange a quick chat with the firewalls guy than have your vendor team stuck for half a day unable to test sending Internet mail. In most cases, it is better to appoint a dedicated "fixer" as liaison for the vendor team than to have them try to resolve support issues directly with your company's usual channels. At the very least, the "fixer" can lodge support requests on the team's behalf, and may be able to provide direct results if the project is time-critical.

#### ON-SITE VS. WORKING FROM VENDOR OFFICES

There is one policy which you should absolutely attempt to enforce which is guaranteed to be unpopular with the vendor. That is the insistence that work be performed on-site, and that assigned vendor team members will be required to work on-site unless specifically exempted by mutual agreement. Obviously you are not going to insist that sick days, etc. be verified with your own firm's manager, but you should make it clear to

Make sure that someone from your team is always available to the vendor team so that he or she can . . . generally grease the wheels.

## Insist on regular reports from the vendor team.

the vendor manager or team lead that any absences need to be reported and also scheduled in advance if at all possible. That means the weekly report should include upcoming absences such as holidays, scheduled vacations, etc. of the vendor team members. If a team member has an unscheduled absence, it should be reported via email from the vendor manager or team lead on the same day.

This may seem like pure old-fashioned control-freak business thinking. Unfortunately, speaking as someone who has worked on both sides of the fence, I find that it is a key factor in a successful outcome when working with vendor teams. While vendor professional-services groups look great on paper, the fact is that the majority of them are extremely short-handed. When working out of their normal offices, vendor team members are subject to phone calls about past projects, visits by sales managers trying to get help closing deals with “just a quick conference call,” deployment managers asking about availability for the next project, fellow team members with a problem that needs solving on another project, etc. Of course, all of these people have pagers and cell-phones, but the degree to which they respond to non-urgent messages delivered remotely is vastly different from the response to someone “just dropping by.”

I have seen most of a day be spent handling other projects’ needs or administrative issues, and the norm is generally two to three hours out of an eight-hour workday. Many vendor team members are very conscientious and will not charge those hours to your project. I have seen many others who will just cheerfully write down “8 hours” for that day, even though most of them were not hours that assisted your project in any way. If the team member is an employee of the vendor, he or she is probably subject to a performance criteria that includes 35–40 hours of billable (revenue) time per week. If he or she is a subcontracted consultant, he or she is doing vendor-related work during those interruptions and certainly has little motivation to donate hours to your firm.

The flip side of this issue is that there is generally great utility in having one or more of your team members working out of their home office for a half-day or a day per week. They will be meeting with coworkers to discuss solving your problems, going down the hall to visit engineering with questions, and basically interrupting other people to help your project. Yes, I understand that is slightly hypocritical. So on the basic theory of commutative benefit in a business situation, it is in your interest to let folks leave after lunchtime one day a week to work from the home office or designate a day that they will always be working there. I do stress that you will almost certainly lose control of the project if you allow the vendor team to set its own hours and on-site policies, as they will choose to work remotely most of the time and communicate solely through architecture documents and occasional emails.

### STATUS REPORTING

Insist on regular reports from the vendor team lead or project manager, as well as brief status reports for the global group meetings. The weekly status reports do not need to be lengthy, but they should follow a standardized format which includes the following:

- Reporting period
- Issues hotlist
- Resolution of last report’s issues
- Team members on project
- Hours billed to SOW
- High-level goals accomplished during this period
- Planned work done during period (by team member)
- Unplanned work done during period (by team member)

- Goals for next period
- Change control information (cumulative, but old info can be pruned if it exceeds a page)
- Updated copy of any project plans or timelines maintained by vendor team
- Scheduled absences (by team member)

#### **MULTIPLE VENDOR TEAMS**

If you are implementing one of the complex service clusters alluded to in the beginning of this article, it is very likely that you are dealing with multiple vendor teams, ranging from your layer 2/3 switching and load-balancing vendor to your layered application-server-package vendor and everyone in between. I feel for you. I truly do.

In addition to the tips and techniques listed here, I strongly recommend that you arrange regular team meetings among the multiple vendor teams, with the technical lead for each major product and the vendor team project manager from each team participating. This works best if you can get overall authority for facilitating, but also works if your boss is a good facilitator. These people may be attending a regular staff or subgroup meeting with your team anyway, but that is not a forum where you can drill deep on product interactions.

Be sure to build verifiable, quantitative functionality test suites into the SOW as a deliverable for each vendor team. When problems arise, insist that each team repeat the simple functionality table and show results. Don't take "we didn't change anything, it should all still work" for an answer.

I know of one major vendor who began building its own vendor services team last year simply because it was losing too much engineering time resolving finger-pointing by other vendor services teams. Since this firm had no team of its own, there was no one on-site to point back and issues were escalating needlessly into engineering to prove that its product was not at fault before other vendor's teams would initiate diagnostics. Be ruthless about the teams doing diagnostics and functionality tests in parallel when the project hits a major snag. You are the client. You are paying for this, and it needs to be resolved.

#### **Conclusion**

A complex service rollout involving one or more vendor services teams is usually a wild ride, but it can be one of the most exhilarating and satisfying experiences found in the workplace. Rigorous application of some common-sense rules can result in a project that ends with everyone feeling good about the project and looking forward to working with one another again someday. Letting things "just happen" and assuming that things will proceed as well as they do in ordinary projects is almost a guarantee of problems somewhere along the line. With knowledge of the workings of the vendor services process and some preparedness in setting up the engagement, you can greatly increase your chances of a fully successful outcome.

Letting things "just happen" and assuming that things will proceed as well as they do in ordinary projects is almost a guarantee of problems.