

How Outsource Software Developers Fail Their Clients

- *an insiders view*

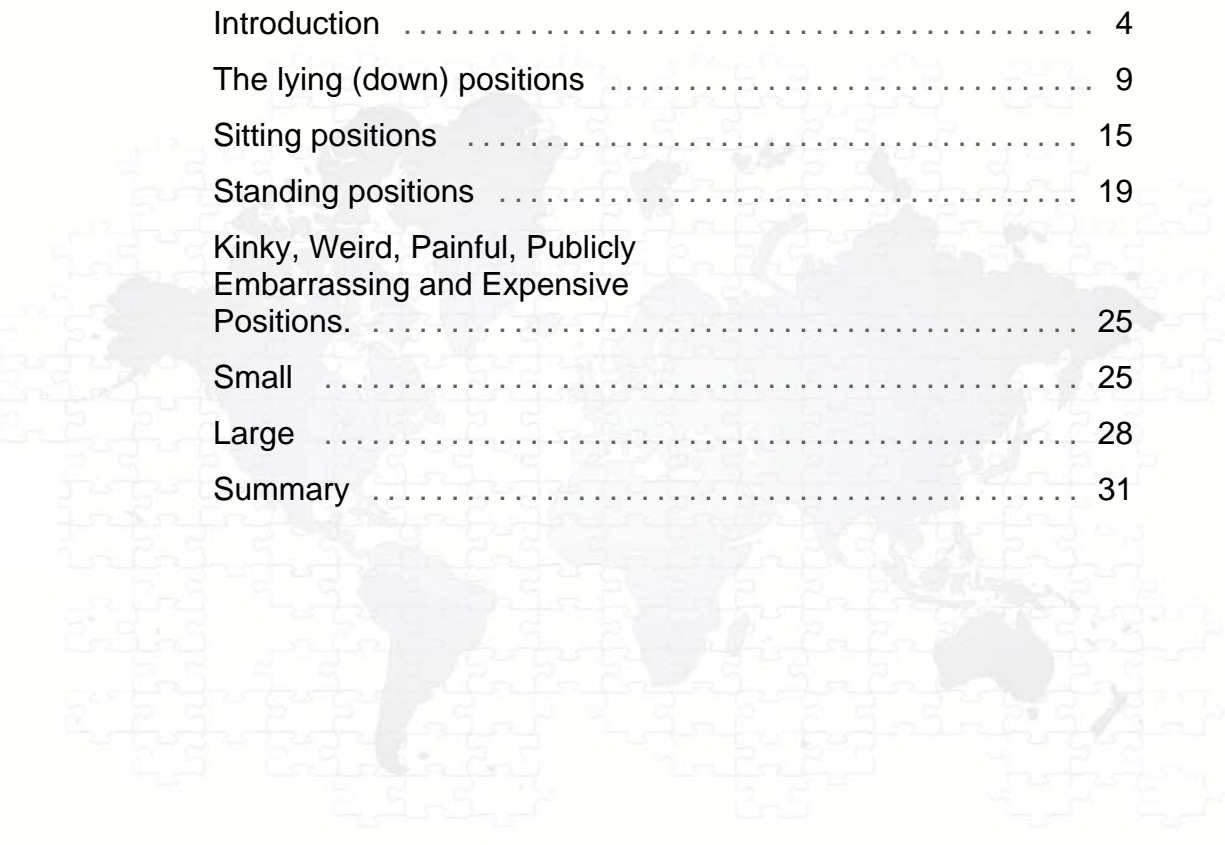


The Kama Sutra

of software development

SOFTWARE EXPERTS
18813 TRACER DRIVE
LUTZ FL 33549 © 2006

<http://www.swxperts.com>



Overview	1
Introduction	4
The lying (down) positions	9
Sitting positions	15
Standing positions	19
Kinky, Weird, Painful, Publicly Embarrassing and Expensive Positions.	25
Small	25
Large	28
Summary	31

Overview

This paper is about how some software outsourcing vendors go about business and how buyers lose. Vendors aren't going to change just because I reveal the rules. To be fair to the vendors, there is no grand conspiracy. That's because the rules they operate under are not written. The rules aren't taught to software developers in school. They are simply a stimulus and response, an action and reaction.

If that doesn't make sense to you, let's use an analogy -- used cars. The rules of car dealing are, in fact, as old as horse trading. If you know the game the used car dealers play, you can often do very well buying a used car. Dealers won't change the rules just because you know them.

That is a real advantage for you, but knowing the rules doesn't mean you won't ever get a bad deal. Even the best can pay less than Blue Book and still buy a lemon. But the odds are much better for getting a good deal, knowing how the game is played.

In overly simple terms: To take advantage of the benefits of outsourcing -- be prepared. Even when it works -- it's hard work.

But like all business transactions there are trade-offs. Outsourcing software is about business. It is about money; **mostly it's about your money.** To make outsource software development work, doesn't take a 300 page book, charts and graphs. Mostly all that is required is common sense. Success is often accompanied with adherence to the fundamentals of business. The problem is often in dealing internationally, the rules don't bend quite like they do in dealing in the US exclusively.

Many (ourselves included¹) have explored the mechanics, policies, procedures and metrics of successful outsourcing. Compliance to best practices is a must! REPEAT, YOU NEED TO EMPLOY ALL THE TOOLS OF BEST PRACTICES!

In most transactions, the catalyst which works best is managing the human aspects. This is no different from doing business in the US, except the partners are overseas, and most do not

¹ Process and procedure papers are available at www.SWXPPTS.com. We strongly support using best practices of monitoring, reporting and oversight.

understand US standard business practices, or our interpretation of the language.

Add the problem that many outsource program shops don't keep statistics, or if they do the management metrics are inconsistent even within a single company. Management skills vary widely from vendor to vendor. So looking at coding skill as the only determinant of qualification to award your project can be very deceptive.

Serious business people know statistics and reports are a requirement to properly manage a business. When vendors fail to keep metrics you lose. Why? Because they don't know how much profit is in a given project. But they will figure it out when the income exceeds the expenses. A vendor's management skills is singularly the most important factor in success or failure in outsourcing. The unfortunate thing is when doing business with an outsource company is this: you may hire great coders but truly bad business people. ²

In the US or overseas, when conducting business, people rely on two fundamental strategies. Buyers and sellers both get less than they bargained for when they don't know which strategy the other party is using. Most often it happens when unsophisticated business people change strategies during the business relationship. These strategies, their telltale signs and consequences are what will be explored in this paper.

To succeed by employing offshoring or outsourcing it helps to appreciate that the software developers are not intentionally out to "take you." Nor are buyers out to railroad the vendor. Yet, both lose when they fail to properly set up and manage the relationship.

Knowing the rules of business and understand how to deal with people you probably already know how to have a great software outsourcing experience. For those who don't know: The key in all business transactions is balancing risk and reward. That is called MITIGATION. The best way to mitigate business risk is by managing the relationship, not the contract.

The methods for doing this are not secrets and they scream out, but only to those who can identify them.

² We at Off Software Experts (swxperts.com) know who uses metrics and we help them interpret their reports. We partner with them for success so your projects will be on time, on budget, and will work.

Introduction

Programmers are a special class of people. For those who don't understand how to deal with or enjoy dealing with people, an alternative to 'an active social style' is 'logical' work. Software, computer hardware and accounting are examples of logical work. Dealing in logic is often a way to minimize confusing or uncomfortable social situations. Software engineers are software engineers because programs are logical. The best programmers are rarely known for being socially agile, rather they are known for their coding and problem solving skill. That is a good thing. Not all programmers are awkward socially, but more often than not they are, thus the name "Geek".

GEEK:

- an expert in computers (a term of pride as self-reference, but often considered offensive when used by outsiders).
- a person regarded as foolish, inept, or clumsy.
- a person who is single-minded or accomplished in scientific or technical pursuits but is felt to be socially inept.

In dialog the other day with a colleague about how difficult it is to communicate with programmers some times, my colleague said: "Good programmers have only two friends. They are both imaginary and they don't play well together." We may disagree with this but, as a reasonable generalization, software engineers as a group are not known for being gregarious people!

Programmers' social and business skills are not a tangent item to this paper or to your success in software outsourcing. They are central to it. Why? Because personal interactions affect developer's conduct from start and into post delivery. It affects how people communicate. This really makes sense in general because, for programmers, computers are logical and easy to understand. People on the other hand can seem difficult, demanding, irrational and unpredictable. People can also be emotional ...and frightening.

Now imagine a group of educated and talented people who would rather spend a day in front of a computer solving intellectual problems than with a group of people socializing. They collectively go out to conduct business with people who are

... 3

social, like to be around other people and who are not particularly logical.

This has no consequence until you want them to develop your software application, then it becomes your problem. Simply, they may not have the business and social skills to be capable and reliable to manage your business project or contract.³

You might wonder, *is it possible then, to have a relationship and a successful deployment?* Obviously yes, but it helps to remember and factor for how programmers think or you could get taken to the cleaner. We recommend you use best practices and leverage the power of an intermediary who can 'translate' for you. Someone who can capture the developer's attention: Hire the 500 pound gorilla.⁴

Programmers are, after all, schooled in programming, not business. This explains why programmers often make poor business decisions. There is a lot more to it than that, but the biggest part is here. Like the parent in the example below, programmers do things without thinking and without considering the consequences. It may not be intentional, but that is not relevant to you, because intent doesn't change things when you are paying for something and not getting it.

There is a rule in life. People will always make the best decisions they can, with the information they have, and their understanding of the situation when they make that decision. - ALWAYS.

Unfortunately, the perception or understanding of what is in their best interest is often defective. What does this mean? It means that people will often base their decisions on faulty information or false assumptions.

Said in another way: People always learn from their mistakes, too often people learn to repeat them. You probably know people who pick bad partners in their personal life. Right? Ever wonder why do they repeat this over and again? Does it make sense to think business will be different?

³ Software Experts assists the vendor manage the contract.

⁴ Off Shore Xperts (www.swxperts.com) has US resident representatives of top rated off shore developers. We provide intermediation and communications support to software buyers.

Here is a rather tragic example, studies have shown that adults who were abused as children are very likely to abuse their own children. When abusers are asked: “Did you know it was wrong?” The abuser will always say they knew it was wrong. Yet in the heat of the moment they still whack their children. Why do they do this? It makes no sense.

The psychological or official term for this repeating bad behavior is ‘cognitive dissonance’ and it is a very real factor governing every business relationship.

Cognitive dissonance is a psychological phenomenon which refers to the discomfort felt at a discrepancy between what you already know or believe, and new information or interpretation. It therefore occurs when there is a need to accommodate new ideas, and it may be necessary for it to develop so that we become “open” to them.

Beyond this benign if uncomfortable aspect, however, dissonance can go “over the top”, leading to two interesting side-effects for learning and change:

- ***if someone is called upon to learn something which contradicts what they already think they know — particularly if they are committed to that prior knowledge — they are likely to resist the new learning.***
- ***and—counter-intuitively, perhaps—if learning something has been difficult, uncomfortable, or even humiliating enough, people are less likely to concede that the content of what has been learned is useless, pointless or valueless. To do so would be to admit that one has been “had”, or “conned”.***

Unfortunately, there is no way of knowing the person who is conducting business has these private habits or whether they will transfer their personal failures to their business transactions. We can’t ask our outsource vendor: “Do you abuse your children? Will you clean me out on this deal?”

The obvious question is: “Why do people do things which are self harmful?” People do things which are harmful because they don’t think; they simply react to stimulus. We would all agree it doesn’t make sense to abuse children, yet people do it.

How does this relate to running a successful outsource project? Software vendors are people. They will do exactly what they believe will be in their own best interest at that moment.

Let's see how cognitive dissonance plays out in the many positions of Kama Sutra of software outsourcing.



... 6



The Lying Positions

The lying positions

Now let's be honest, people lie. In the outsourcing business, most people don't intentionally lie, well not more or less than in any other business transaction. It just seems that way. In reality it just goes back to making decisions and saying things which seem the truth at the time. Almost all software developers are honest, hardworking folks (*who would never beat their children*). Except... ..well except they didn't think about the consequences when they promised 200 hours of work on a project to be done in 5 days. *It's just a little lie, right?*

For the remainder of this report the buyer will be known as **Kerry Oki**, and the vendor as **Dementia Software**.

Kerry Oki is an office manager at the temp help firm Labor Pane. She needs a Customer Relations Management system (CRM) which the firm believes is resalable to others in the franchise.

Dementia Software is overseas with 22 coders. It is run by Bi Polar a tech graduate from a local university. The rates are \$14/hour. The programmers are all committed and the clients should be.

... 8

What happens is this. Kerry Oki, the buyer, signs up with a vendor, Dementia Software for say a \$10,000 test project. Half way through, deliveries start slipping. Kerry Oki starts to press with emails and calls. But the deliveries keep slipping. The long and short is Kerry leaves with the project weeks after it was due, and 1/3 done. She agrees to pay half, or \$5,000, just to cut losses. Dementia Software sees it as buyer demanding more than they has originally agreed.

Kerry Oki blames Dementia Software -- she got taken -- and Dementia Software likely will blame Kerry.

See the reward? No? Dementia Software got paid. Getting paid is positive reinforcement, and therefore the propensity to repeat is nurtured?

It may very well be Dementia's fault. But given the situation it is likely the next buyer will experience the same thing in dealing with Dementia because the bad behavior was reinforced.

All will agree that Dementia Software didn't deliver as promised. But could it be Kerry set herself up for this little intimate tryst between Dementia Software and her pocketbook?

In reality it just might be about Kerry Oki's behavior. Even if it is an equally shared responsibility, the blame belongs to Kerry because it's her money and her responsible for intelligently protecting her assets.

Agreed, this is an insider's view. Again, everyone learns from their mistakes. Dementia Software has learned to repeat theirs and because of that the next buyer will likely be viewed as a teller machine to kick if it doesn't give change. One third the work and one half the money is not a bad deal for Dementia.

But what did Kerry learn?

Well, this is complex, but if you think like a (new) buyer you will see the other side of this. Buyers often use price as a primary determinant.

Kerry looked at the sample work of multiple vendors and she selected Dementia Software because the standard was acceptable. Further, she did the research and confirmed the competency of the Bi's team. So Kerry did her homework and perhaps trimmed it to five who might have made it to the short list. All five could do the job.

But here is the trap. All buyers always make their decision with cost as a factor. No one makes significant decisions without considering cost Kerry Oki picked a team at \$10,000 in the lowest 35% of the bids with the median at \$8,700 and the highest at \$18,800.

With all the pieces to the puzzle on the table let's put them together and see what we have:

- ✓ Kerry did the homework
- ✓ Dementia showed competency
- ✓ Price was low
- ✓ Deadlines reasonable
- ✓ References checked
- ✓ Delivery schedule, milestones, payment schedule and SOW agreed

Let's assume they follow good business practices and the following are also in place:

- ✓ There is a bug tracking system
- ✓ There is a progress monitoring system

✓ There is a QA monitoring system

So what could go wrong? Kerry did her homework. All the angles are covered. Right?

Unfortunately, lots of things can go wrong. The most prevalent one is likely simply reactive thinking of Dementia. It usually goes like this:

I need this job. Got to pay the bills. I have to cut pricing to get this job. OK, we can do this job. The C# team will be done with the CRM project on Tuesday. And, hmmm, it will take us 4 man months. So at \$15/hour that's \$9,600 and allowing for margin we will round to \$10,000.

OK I will bid \$10,000. Should be easy.

Bid is tendered and accepted by Kerry.

See the trap? Can you see where the project will derail?

No? Well, it is centered in Dementia Software management. Usually this is not visible to the buyer. In reality only the best managed developers can consistently make deadline with certainty⁵. Whether you know it or not -- there is an unwritten law of software development. **“It will never be done on time.”**

It's the law! Software will always be late!

If I were a gambling man I could make money giving odds that Dementia's current project won't be done on time. Even if properly managed companies know there is always post delivery cleanup and bugs. The best run companies know and incorporate risk management in their proposal. Few programmers know how to manage risk effectively. But Kerry, like many buyers, is not skilled in assessing her risk. She is unable to discriminate between simply a higher high price (greed) and price with risk insurance. Dementia, like others who often will cut price, the risk or the quality to get the job. Kerry should understand that increased risk raises the price, but how can she tell which is which? And in software development, quality can be a very hard to predict for a novice buyer.

Going back to Dementia, what no one outside the Dementia knows is whether the current CRM project is weeks behind already or not. Kerry has no idea of what is going on inside Dementia Software.

⁵ Software Experts works hard to monitor our vendors and by verifying their management standards. We will not work with those who violate delivery schedules or are not responsive to our buyers. Additionally our vendors will provide a performance and delivery guarantee.

So Dementia Software takes Kerry Oki's contract, whether the current project is late or not at this time. So what strategy will Dementia employ to kick off Kerry's project?

We can be reasonably confident the existing project is late, though it really doesn't matter. Here is why. Dementia Software robs the current CRM project to satisfy Kerry's new project.

Why would he do that?

To understand, peek at this from Dementia's view. Bi Polar has to protect his business. Kerry's project must start without a hitch. Dementia meets payroll with Kerry's deposit and cycle continues.

And poor Kerry thought it would be frankincense and myrrh.

For argument's sake, suppose the existing project is done on time when Kerry's comes up and it is 2 man months. Dementia knows sales must continue unabated, so unaware to Kerry, two or three weeks into Kerry's development, Dementia takes a deposit on another C# project. Kerry's isn't done yet but the profit margin on the next job is higher than her's. So Dementia robs Kerry's team to start work on new project. Now both projects will be late.

This is not the only way it works as there are variations of this theme. Here are two more examples.

The lead PM is off on holiday or vacation and the scheduler forgot to tell the boss or the boss forgot to inform Kerry when they accepted her project. Or...

This is the worst of the lying positions.

Dementia doesn't have the skill to complete the project. End game -- Check mate. No matter how much energy Dementia puts in to Kerry's project, they won't ever be able to make it work because they haven't the skill! They will milk Kerry for as long as they can dreaming that the skill will miraculously appear with Elvis dressed and the tooth fairy.

YES! Kerry Oki loses big time as in BIG. But there is nothing exotic about this. All these are just plain vanilla variations of the lying positions in the Kama Sutra of software development.⁶

⁶ We at Software Experts know these things. We help our buyers find companies who don't play these games. We roll up our sleeves and get in the ring with our clients.



The Sitting Positions

Sitting positions

Let's leave the mundane behind us and go for something just a little exotic. Not anything wierd, mind you, just not the standard old pabulum.

But before we explore new exciting ways of getting taken to the cleaner while sitting up and paying lets go back and review 'cognitive dissonance'?

and—counter-intuitively, perhaps—if learning something has been difficult, uncomfortable, or even humiliating enough, people are less likely to concede that the content of what has been learned is useless, pointless or valueless. To do so would be to admit that one has been “had”, or “conned”.

Now you say: “Sure, Kerry messed up. She took a bad vendor. What’s the problem? Are you suggesting Kerry is at fault?”

Kerry Oki agreed to accept a bid at \$10,000. The SOW and schedule are in her hands. Kerry sent 1/3 deposit on a project for delivery in 60 days. Everyone is pleased. The instant messages start flying and the questions come. The first week it is “Happy, happy! Joy, joy!” Kerry goes back to the boss and says: “Hey boss, we are on track. These guys are awesome! Look at this code they delivered in just 2 weeks! This is GREAT. Great, I tell you!”

This is starting to look like a love match made in heaven. Nirvana is so close she can touch it. Ecstasy and a whole new level of excitement are just a couple weeks away!

In the third or perhaps the fourth week of hearing nothing Kerry thinks: “These guys are working day and night. I don’t want to interrupt them. I would be a distraction! I don’t want to slow them down. I saw their initial work and, well, they can do a stellar job.”

So she waits. Still there is only the sound of silence. Finally she asks Bi Polar: “What’s happening?” The answer is as predictable as wet on water. “No problem. We are busy on your project. No problem! We are doing fine. Delivery will be on schedule.” And that is it.

Dementia Software sends no details, just a pleasant missive which renders to, “shut up, and go away.”

Kerry thinks: “Well OK, they are working.” Now it’s the fourth week and a milestone is approaching. ‘Happy, happy: Joy, joy’ has been replaced with Kerry in the office of her supervisor. She is saying: “Look, I called the guys last week and they said it would be ready yesterday. I sent a message but

... 13

they haven't responded yet. I will get on the phone (Skype or IM) tonight and see what is going on."

Her boss tells Kerry to get on it. Or perhaps he tells her to **'get on it -- or else!'**

Kerry Oki is concerned! She gets on her primary communication channel to pressure the vendor. Rightfully so. Bi Polar apologizes and tells her there is a minor problem. See the PM or lead coder was out sick. Perhaps the system was down, or some emergency arose. Maybe there was a holiday or there was a system crash. Perhaps they changed a new rev on the OS. Doesn't matter, take your pick. The point is that the excuse is plausible!

The promise from Dementia Software is delivery will be shortly. They tell her that her patience is truly appreciated. But she knows things are not quite where they should be, but there is nothing she can put her finger on. There is nothing definitive other than the delay.

Kerry wants to believe them. She has invested in this. Her reputation is at stake. She desperately needs this to work! But guess what? There is another sound of silence hitting like a hurricane.

Now it looks like an even later delivery on the milestone and it is now 4 weeks late and it turns to 6 weeks late. Dementia request some piece of information, perhaps something in the specifications not clarified in the opening Statement of Work (SOW). They say it wasn't clear or some other such excuse. They are preparing for making their failure to deliver Kerry's fault. They are looking for excuses. They have done this before, they are good at it. Kerry is not. Dementia is at the advantage.

By now she is very concerned and feeling discomfort. She knows her boss will be pressing her for delivery. She is seriously concerned!

Review, cognitive dissonance:

the discomfort felt at a discrepancy between what you already know or believe, and new information or interpretation. It therefore occurs when there is a need to accommodate new ideas, and it may be necessary for it to develop so that we become "open" to them.

"NO! NO!" She thinks. *"I am an experienced business person. I am not subject cognitive dissonance! I DO NOT BEAT MY CHILDREN!"*

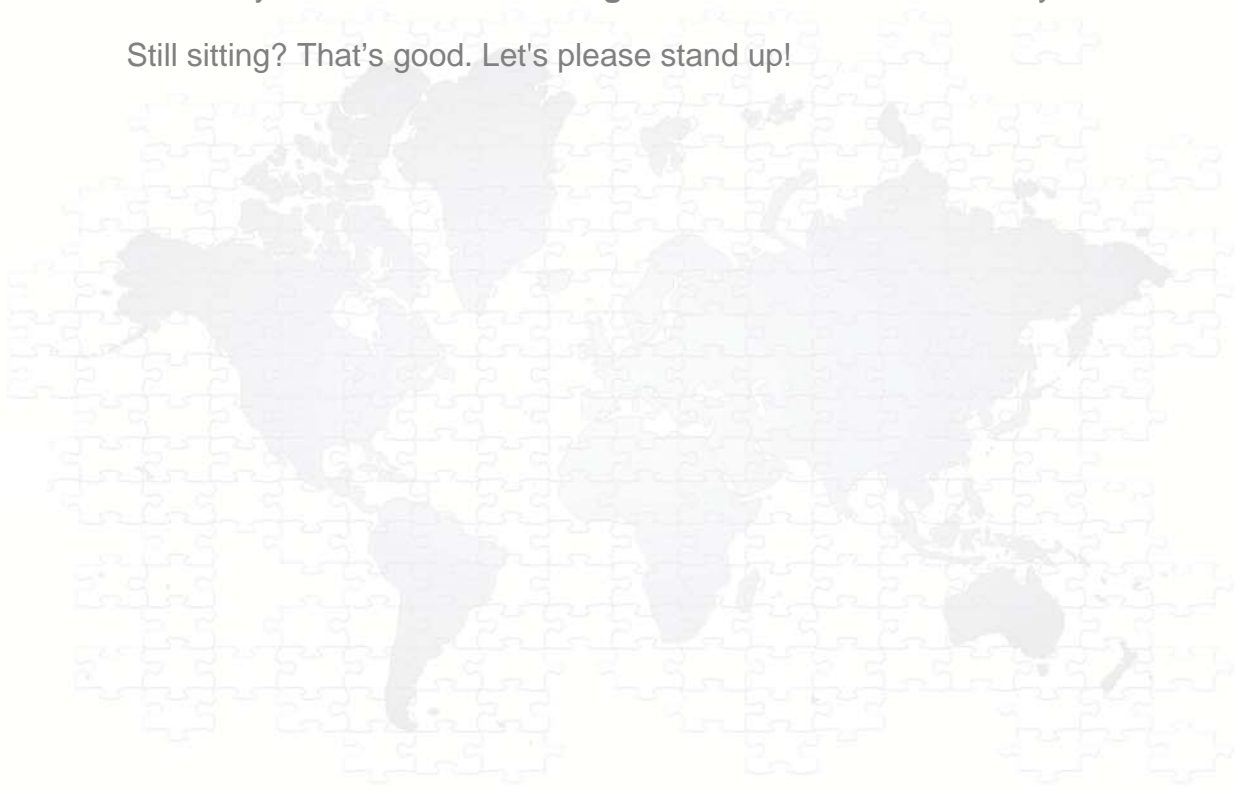
... 14

Her problem is the project is now months behind schedule, maybe over budget also. And she still hasn't a product to deliver to her clients. So what does she do? ⁷

Most Kerrys will bend. They will accept terms and conditions not dreamed of when proposals were first reviewed. For most Kerry(s) it may be the first of their life in a dysfunctional relationship (business or otherwise) and they don't know how to get out.

What they don't know is: **It won't get better.** No matter what they do!⁸

Still sitting? That's good. Let's please stand up!



... 15

⁷ Find the best solution at the conclusion of this paper. Software Experts offers alternatives to this scenario if you begin the process correctly. It is very difficult to clean up after the fact.

⁸ There are options, but only if you start properly and insure your risk. Let Software Experts show you how.



The Standing Positions

Standing positions

The elegance of the standing positions are so exquisite that often the client doesn't feel a thing, see a thing or know a thing. It's as smooth as a professional pick pocket and as graceful as a disappearing act by a professional magician. Sometimes it goes unnoticed for years. Other times the client never knows -- not ever.

Sounds impossible? Well, that's because the standing positions are often lubricated by the client's demand for a very low price. It's called GREED.

In a perverse way these are beautiful in simplicity! Ravishing in execution! That is, unless it happens to you. Note: These are not hypothetical, they really happen, every day.

These will be in a slightly different presentation so you can see it, first the set up and then the switch:

Set up #1: Kerry Oki has bid the same job for \$10,000. And well, surprisingly everything goes as planned. They meet the deadlines and benchmarks. All goes well, there are some bugs and technical problems but overall she goes live only perhaps 30 days late. All is happy. Happy! It rarely gets better!

Dementia Software gets paid and Kerry goes live. You should ask: *"Where is the problem?"*

Set up #2: Kerry Oki has a profitable business which uses the web. She sees a competitor 'Scar Tissue' which has some features for federal reporting -- "Section C Reporting" -- she wants for 'Labor Pane.' She does a public record search and knows Scar's business isn't doing so well. She does more research and finds out it will take 3 months and \$30,000 to develop this feature. She sees it working and offers Scar Tissue \$50,000 to buy it. In the deal she would get the domain, the software and client base. Both parties accept. All is happy. Happy!

Again you ask: *"Where is the problem?"*

Set up #3: The new site Kerry Oki set up last year is working like a charm. Kerry Oki is in labor and making money. The site has some problems but she's dealing with it. But unfortunately, her lead programmer quits just when she needed to add a Real Simple Syndication (RSS) feed.

Those are the set ups. Now we can explore the switch:

... 18

This is where the whole thing comes together or more appropriately comes apart -- call it 'the awakening.' Let's start with #1 and see how Kerry got cleaned in the three most common variants of the #1 standing position:

Standing #1a. There was no problem until Kerry wanted to make changes to have the software modified. Then she discovers that there is no documentation. That is actually not unusual. Low cost or on-the-job programmers often are not formally trained, and even if they are writing is not their favorite activity.⁹ Add to the human factor that English is often not the first language. As a result most off shore documentation, which is the developer's notes annotations used for navigation in future modifications to the code, are just missing or poorly done. Well sometimes there really is good and accurate annotations but that doesn't necessarily change things, much. *I actually saw code which was very well documented in, don't laugh now, Russian!*

A lack of documentation is much more common than you might believe.

#1b. In a variation of #1a the code is just fine and everyone is happy. Things are going along well and perhaps a year later Kerry decided to change hosting servers. She ported her site to a new host and everything crashes. The site doesn't work when moved to the new servers.

She immediately called M. Ellen Cally at her Internet Service Provider (ISP) host to light a fire. M. Ellen says: *"The hardware is fine! It's the software."* In panic she calls her contact Luna Tsee, at Dementia Software to discover the development team is busy, disbanded or otherwise not available. But if she is willing to drop say \$35/hour premium rush, Luna will have someone look into it. Kerry agrees 'yes' and they say: *"2 days minimum in advance."* She wires \$560 immediately. It takes 3 days to get there.

So it's close to a week later, Dementia Software has the money. They go into Kerry's site and do an analysis. They come back and say; *"Nope! Not us. It's the ISP."* She believes them because she just paid \$560. But she really doesn't care who is at fault -- she wants it fixed NOW!!! So she goes back to the ISP to tell the ISP to fix it. They say: *"Nope! Not us! It's the program"* or -- You will love this, *"You have a memory leak. Go back to the programmer."*

Extremely frustrated and almost out of control with this dance of technicians, she finds someone with system administration skills to look at it. It's then she discovered the original programmer called some unusual operating system configuration file in the initial set up. Some feature perhaps in an old version which needs to be working and wasn't installed

⁹ Off Shore experts insists our vendors have degrees and many have advanced degrees in software engineering. All are required to submit samples for compliance standard review.

when she moved. An old version of microsoft something. You get the idea. *Was it the programmer? Was it the ISP? Was it her? She is relieved.*

#1c is perhaps the best of this set. The code works well. Kerry Oki can port it, it's well documented, she had it modified for new features and everyone is happy! She went to work on Monday morning and discovered her site had been down since Saturday. Many transactions were lost. If she is lucky there are hundreds of emails advising her of such. If unlucky, she hasn't a clue till she logged in. In a panic she calls M. Ellen Colly at the ISP to get the site re-booted. It comes up and works fine. Everything is happy again! She asks around and everyone says: *"Well sometimes things happen."*

A couple weeks later it crashes again. This time perhaps, if she is in luck it was discovered before much of a loss. Justifiably, still she is concerned. The ISP can't find anything and her tech team can't either. It's a fluke, but it happened more than once. Then it happens again. She calls the ISP -- and guess what? M. Ellen goes: *"Nope! Not us. It's the program"* or *"You have a memory leak. Go back to the programmer."*

You can bank on the programmer saying: *"Nope! Not us. It's the ISP."* So you go back to the ISP and tell them to fix it.

To make short of a very long gestation for problem resolution it turns out there is a design flaw in the original specifications. Whenever Labor Pane has more than 250 concurrent users the system reaches capacity and locks up. She needs a redesign of the entire system for load sharing. She didn't specify the load balancing in the original specs and the original team didn't advise her to consider it because it would have added 20% to the project. Suddenly now, she remembers the bids at 20% higher which she turned down, \$12,000 ~\$15,000.

The other developers on the short list said: *"You can pay now or you can pay later. But long term, it costs less for us to do it right!"*¹⁰ And Kerry discounted them because she didn't understand how to discriminate between a bargain, a value and a bad choice. Now she will have the privilege of paying for two deployments of the same system. The one she paid for and the one she will get to pay for.

Who is at fault? It doesn't matter. Kerry will pay for it regardless who is at fault. ***Ouch!***

In example #2 Kerry, the buyer bought the code. They brought the application to Dementia Software to be integrated into their existing

¹⁰There is an invisible line that exists when people bid their jobs. The line is between paying just a little more and getting value and paying just a little less and getting taken. It is very hard to see when you enter the process. Contact SWXPERTS.com before you bid your job and let us reduce the risk.

business. The problem in building out the code is frequently exacerbated by bad code techniques.

It did NOT go well. For this example we will introduce a new vendor Quick2Code (an alias) who is the good guy and one of our vendor partners. Like all other examples, this really happened.

First, a bit of background. Many off shore developers are not formally trained. They don't employ standard or even minimal code quality practices. This can be explained by the example of a simple 'call.' In good programming technique, when a programmer has to query a database (DB) they have a couple of choices. The programmer can write the query and put it into a table. Then every time they need to query the DB they just insert the instruction 'call DB query.' They reuse the same block of code but simplify the application by use of a shortcut. The main program has a 'call DB query' in it.

In poorly written applications the query is imbedded into the program itself. Thus if a program queried the DB 100 times the same block of code is repeated 100 times.

Both programs work! One is substantially more efficient and will run faster with fewer problems than the other.

The slight of hand is here. The difference in original cost might be \$1000 (10%) Kerry couldn't tell the difference between clean code and bad code when she bid the job. When she needs to change the program query section, one case would require finding and fixing all 100 iterations rather than just make the change to one block. Smooth isn't it? Kerry would never know she was cheated until she took the application to someone else.

For Kerry Oki, the entire code needed to be rewritten to fix the problem because it was functionally unusable -- the maintenance costs and support would have life cycle costs which would be excessive. Our vendor, Quick2code refused to accept the work¹¹ because they knew Kerry would never be satisfied. If a change was made in one location in the code it could easily cause a problem somewhere else. Kerry was definitely not happy. In fact she refused to have it rewritten and pulled the job from Quick2code and gave it to Dementia Software to patch.

What makes this truly interesting is she confirmed with three other vendors who advised her that it was not economical to patch. Yet, call it greed, limited budget, competitive pressures or cognitive dissonance, she made her decision to select a company who would patch the existing code.

Kerry displayed a poor choice when she purchased the code without analysis first, and then followed that up with an even worse decision. If

¹¹Our vendors will not take projects which have very high risk. In this case it needed to be rewritten, and the client refused to pay for that.

there is a classic example of a buyer with cognitive dissonance this would be it.

The #3a switch example provides a very interesting situation. Her in-house developer left. He was good and kept things doing well for the time he was there.

So Kerry did the most expedient thing, she took the code and bid a job to add (RSS) to her site. She got some low bids and approved the Non Disclosure Agreement (NDA). Half the folks responding say it will cost you \$10,000 and the rest say: "Oh yes, we can add RSS for \$1,000" The discrepancy is shocking. What? Why the big gap?

Well programmers can be curious people sometimes. But the reality is in the mindset of the ex-employee. In developing Kerry Oki's code the employee built in his own job security. The annotations were cryptic, which is not uncommon by the way, or some other method of securing his continued employment. It's called job security.

In a **variant #3b**, the new vendors will look at the code. They see it was written in an early release of VB, or perl. Today's technology is way past that and to patch this software will have the long term impact of trying to run a windows 3.11 application on Windows XP 64 bit. It may be done but it's not a good idea.

The standing positions are real but it does get worse!

Now we will enter the twilight zone of the Kama Sutra, as you will discover very few can go here.

... 22



**Kinky, Weird, Painful, Publicly Embarrassing
and Expensive Positions**

Kinky, Weird, Painful, Publicly Embarrassing and Expensive Positions.

You DON'T ever want to go here.

In general, these positions have to do with intellectual property (IP) rights. IP issues will generally fall into two categories: Small and Large.

Small

The small ones are the ones you need to be concerned about. Here is how people lose IP in these positions. These positions are only possible in the twilight zone of buyer's cognitive dissonance. That means what clients say and what they do are in direct conflict with each other.

Here is the classic setup. Kerry Oki is planning to package her new labor CRM as "**Delivery Distress**" (DD) as a client-server, stand alone system. She wants to introduce it with a variant. This will be an Application Service provider (ASP) on line service model as "**It's Just Temporary -- get over it**" (IJT).

She has \$35,000 for a budget to Beta DD and another \$20,000 for IJT. She has a \$50,000 for marketing before she goes to a venture capitalist. So what does she do?

If she was smart she should go to a lawyer for an NDA to protect the IP?¹² Perhaps, but most Kerrys just rip off some NDA they find on the web or in their files - (A BAD IDEA.) Next she looks for a developer with experience and skills in CRM. She picks developers by looking at their past experience in CRM. That should be good, right? Sure.

Kerry is diligent and will probably look at a long list of possible candidates and will narrow it to a short list of say 10. The top 10 get to sign an NDA before she will let them look at the specs, or perhaps any existing code. Kerry wants to be absolutely sure they aren't going to take her code and design and give it to her competitors. Worse still, because this is the greatest labor delivery management system EVER, she wants to make sure Dementia won't repackage it and sell it against her. Make **sense**? Sure it does!

No, actually it doesn't. To understand why this is felicitous first see it as a form of cognitive dissonance.

She wants to be sure her software remains her software. It should, after all be hers, because she is the one paying for it. The NDA may or may not be

¹²Software Experts has U.S. applicable and mutually protective NDA agreements available to our clients.

abusive. It may or may not be enforceable, but no matter, she will not speak to anyone or discuss anything until she has one signed.

NOTE: NDAs are proper course of business! Absolutely employ one!

But the reality of an NDA, particularly a restrictive NDA, is a bit of foolishness. Let's see how this plays out between logic and reality and where the client, by virtue of her own action, can set herself up for the IP switch, which is the most painful of all.

First, software technology is changing so fast that anything you build today will be obsolete in 5 years, and likely sooner. New languages emerge every year or less. Each to make the development process faster, Java was hot, now Rails is the game. What was HTML is now Web 2.0. Your code has a shelf life, and some would say should come with an expiration date. 64 bit processors and operating systems are just around the corner. 128 bit perhaps 5 years, and 256 bit within your lifetime.

Second how did she pick her vendor? By experience and price. Lets assume she even factored in delivery schedule and it would be reviewed by a third party.

She found a company with the experience building successful CRM and the price is in the lower half of the bids @ \$35,000. The highest being \$102,000. For Kerry it is like magic.

Now here is where Kerry gets a painful dose of umbilical twisting. Her cognitive dissonance creates two worlds which can't possibly exist in the same space-time continuum. Impossible to exist!

Dementia Software is able to build the CRM in the allotted time, they have the experience and knowledge in CRM. *What could be the problem? Here it is - they will make a profit! How do you think they can possibly do that? How can they be in the lowest half, deliver on time, and have a viable application while still making a profit?*

Right! They will reuse code, applets, libraries, macros, etc. from previous CRM development projects. The very same ones she looked at when Kerry checked their experience. Get it? Right! You had better be thinking: *"Oh, gee, I get it!!! I guess that means when I buy an outsource vendor, they will be putting my stuff in their library as well. Well what about my NDA?"*

That is the conflict. Kerry has an NDA and wants experienced partner. See?

Well, that takes us to the next aspect of this dilettantic escapade.

... 25

The final exercise in this set is where Dementia Software has an extensive library of existing CRM applets that others have paid for! Whether or not you accept that your thinking, it is a reality. And Kerry's program development may just reward the developer with new resources in their library, new features and new ideas for their next job. It just might be that Dementia's next client is "Labor Party," a competitor of Kerry.

So Kerry Oki decides to engage a company with no experience and Kerry will pay for their learning curve. It would seem this would protect Kerry along with her NDA. Does it protect her from having her code, applets, macros or the like reused on Dementia Software's next client? No! She gets to pay the price for them to learn creating her 'original code' but when done Dementia is no longer a virgin to CRM. All bets are off.

You are asking: "So, how do I protect myself?" Well, maybe a little common sense will help.

First go to an expert or consultant¹³ because they will know who to trust, but if you go it alone, look at your vendor. Does your vendor have turnkey products they sell? If yes, challenge them. They will likely tell you: "It is good because it means we have experience in this area." A logical and reasonable answer. But not the right one.

An expert will help you balance the risk and reward. That being, if you want security for your application, don't use a firm which sells their own boxed software! Particularly if theirs competes in your genre. The one important exception is when you are knowingly buying a software package and want it customized! For example Kerry could buy an Opensource CRM, pay for the customization if she were to use it exclusively in-house saving herself significant development costs. Or she could customize it, pay the license or royalty fee and resell it as her package, again saving money. Licensing Opensource is one way to cut cost and comply with the license agreement.

Otherwise, when she goes to a firm with her software and they have an existing competitive product, she is giving them a license to steal. It makes no difference whether Dementia's package, 'Stretch Marks' is successfully marketed or not.¹⁴

For them it is the best of all worlds. Kerry gets to pay for the development of her product while Dementia will integrate her ideas into theirs.

¹³This is what Software Experts (swxperts.com) does. We validate the qualifications and risks of vendors.

¹⁴We work closely with our vendors and know the ones you can trust.

Don't think all is doom and gloom. Developers, as well as too many buyers, have this over simplification of what it takes to sell a software application. Very few buyers and even fewer vendors have the marketing muscle to be able to sell their own products. The simple reality is selling boxed software is hard work and very expensive.

So Kerry Oki has an NDA. The problem is this. It is almost completely unenforceable. *Why?*

- ✓ Kerry doesn't have a large enough budget to enforce an NDA. Lawyers and litigation for only US based litigation is expensive. And off shore litigation is simply prohibitive unless she has very deep pockets. It will cost many times Kerry's initial investment and he claim will have to go overseas where the laws may be different. An NDA may not be enforceable even if she has unlimited resources.
- ✓ Kerry doesn't have the inside knowledge of Dementia Software's other clients or others' code. How can she know if Dementia resold her code? The rub here is simply this: How can she possibly enforce an agreement if she can't prove a crime?

Kerry Oki can mitigate this by finding a reputable intermediary or consultant within the US. someone with some power or authority. Ideally with US contract providing her an equal footing should things go wrong. In most cases there is no premium for this but those with US based representation provide lowest risk but are generally not the lowest price.¹⁵

Large

This is where the big boys play, and get hurt. I won't spend much time here because it is rare. The case has been made here that outsourcing software can be a VERY dangerous game! Even the largest, doing all the right things right, still get into compromising positions. The only difference is the pain, cost and size. When the big boys play it is much like dinosaurs mating. Loud and public. Unfortunately many innocents are caught in the whirlpool of activity, not knowing if they are inadvertently party to using 'hot-ware.' As a result it costs as clients will often make alternative and less risky choices.

SCO suit now seeks \$3 billion from IBM
http://news.com.com/2100-1016_3-1017965.html

Microsoft's source code illegally published on web,
http://www.sophos.com/pressoffice/news/articles/2004/02/va_msleak.html

¹⁵Software Experts's clients, have mitigated their risk by dealing only with vendors who have local contracts, phone and representation.

Cisco source code published on the web. - An inside job?
[Http://networks.silicon.com/telecoms/0,39024659,39120745,00.htm](http://networks.silicon.com/telecoms/0,39024659,39120745,00.htm)

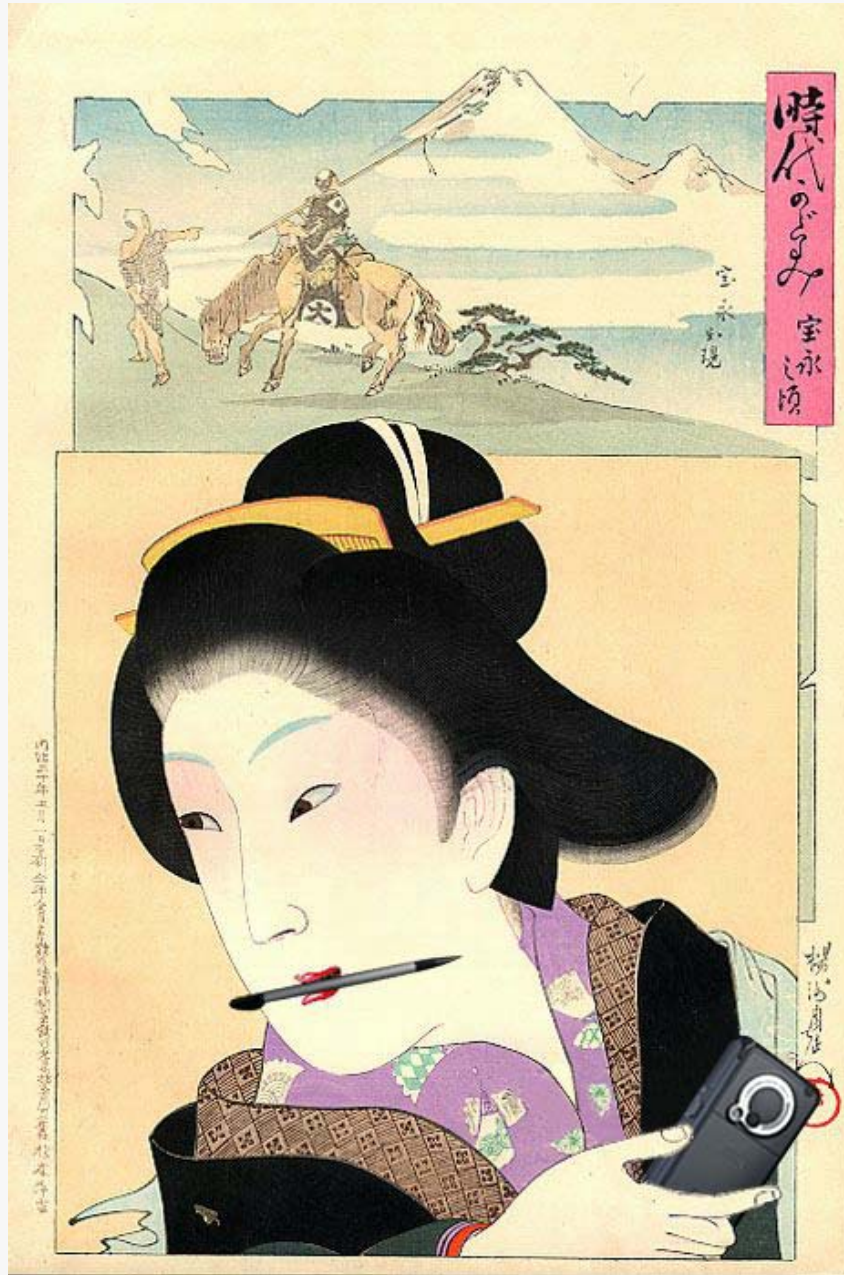
And then there is Intellectual Property Security.... ...a whole different area.
Check back at SWXPERTS.com for new report on security coming January 2007

Yahoo News and Adobe Acrobat:

<http://www.securityfocus.com/news/254>



... 28



Summary

Summary

The advantages to off shore software development are repeated over and over on every outsource vendor's site. There is no need to repeat them here. You have heard war tales and horror stories; Only some are included in this paper. Clearly there are many ways you can lose when outsourcing your software development and your size alone doesn't prevent it.

Knowing and understanding how others have lost can only help you prepare. Fortunately there are things to do which will mitigate your risk. Unfortunately the only way to eliminate all risk is to avoid all activity.

Hopefully this paper made a case for one safe strategy for outsourcing: **Mitigation**. Most of this is common sense, like making sure the vendor has a track record, verification of the skills, checking for references, investigating the communication channels, insisting on a good project tracking systems, and monitoring activity. If something doesn't feel right, it probably isn't. In other words, do your homework and participate. And yes, it is hard work.

You should ask: "Beside just knowing the variations, is there a better way? Is there something else I can do to protect myself?"

Yes, there is. Often it is simply just as easy as finding someone to assist -- an intermediary, a professional. Just as when you hire a Realtor, or a lawyer, you borrow a professional's experience. Don't go it alone. Like a Realtor, Software Experts (www.swxperts.com) doesn't charge the buyer for its services.

Leverage us. We do the homework; we participate; we do the verification checks. We are your 500 pound gorilla. We only represent a limited number of selected global software development firms who meet our standards. Some of our representative companies are young, aggressive and are willing to do things which will build their reputation and reward you with a lower price. Others are established with experience in very complex systems and are busy enough to demand a higher rate. All are willing to provide performance and delivery guarantees. It is our job to help you balance urgency, cost and ROI.

We can guide you through the risk reward maze because we don't 'shop' your job to get the lowest price but will place it only with our affiliate developers who have shown an ability to meet delivery standards.

... 30

Can SWXPERTS.COM vendors provide you with a guarantee? Yes, definitely! You can request a performance or delivery guarantee from our vendors, perhaps even a late penalty if you want. Our vendors accept penalty work at a modest premium say 5% ~15% or with an offsetting performance bonus.

But you are in business and know success comes at a price. Stated differently, you get what you pay for.

All Software Experts' vendors must adhere to a professional standard because we are their major conduit for business income. We are their income stream and that provides substantially more influence on their behavior than any one client alone.

One last thought. By definition; half the programmers in the world are above average, and the other half are below average. Do you know the difference? -- We do!

We at Software Experts do the homework and only deal with those vendors who meet your demanding standards so you don't have to.

It IS that simple.

Call a Software Expert today.

... 31