*## Column: Editor's Gossip Column*

# How To Charge $500/Hour – Part II

*Whil Hentzen*

In the two years I've been at the helm of FoxTalk, no editorial has ignited as large a response as my November, 1997 editorial, "The Starving Artists Club" (although another editorial – the January 1997 epistle about my toenails – came pretty close, albeit in a negative way. But more on toenails another time. *I promise!*) It was rather scary, actually, since, for the first time, I had to write a sequel that people were already anticipating.

## *The problem, circa fall 1997*

To refresh your memory (and mine, since, as you get older, the mind is the second thing to go), I posited a number of reasons for the incongruity of the situation we find ourselves in again and again. This is where an underwhelming, inexperienced youngster can be charged out at ridiculous amounts for routine, commodity-like work, while sophisticated, experienced developers doing serious mission-critical development struggle to justify every dollar they dare to invoice. Clearly, our customers don't look at us in the same way as they look at their accountants, attourneys and architects. They don't perceive us as professionals, and are unwilling to pay in the same manner. They don't feel they receive the same value per dollar. I then suggested that I had an answer to this conundrum.

## *In the meantime*

Something has happened in the interim that's made this problem disappear, making the answer irrelevant. But something uglier has reared it's head.

A front-page article in a recent Wall Street Journal (Business Outlook, Monday, December 1) discussed the difficulties that companies are having finding technical people. No, they weren't talking about problems replacing a friendly smile at the PC help desk. Rather, mission critical projects that were architected to produce a competitive business advantage are starting to get *cancelled* (not postponed – but cancelled!) because of the dearth of talent. Silicon Valley firms are reporting waits of a year to fill technical positions. College graduates who haven't gotten a checking account yet are reporting starting salaries of $70,000.

I think the tables have suddenly turned. I don't think we have to worry about rates, at least for the next couple of years. It's a seller's market, and we're the sellers. We can raise our rates and simply live on the bubble.

Unfortunately, no one else has to worry about rates either. This sudden paucity of talent has spawned a number of scenarios. People are leaving companies to go out on their own to take advantage of the spike in demand. Those that remain are seeing salaries skyrocket, causing hard feeling with peers and stomach acid attacks in the human resource department. As a result, there is a new flood of hacks knocking on doors – those who have never had to collect on a past-due invoice, or deliver a complete solution to a customer. The rest of the company provided the rest of the support necessary.

This new breed of hacks can wear the fancy $1,000 suits, throw buzzwords from the latest InfoWorld around, and easily snow two large groups of customers. The first group is the unsophisticated, small company folk who may not have ever bought custom software before, and while not stupid, are often naïve. The second group is the experienced, dyed-in-the-wool MIS coterie who have done big systems for decades, albeit not on a PC platform.

A rising tide lifts all boats – both those seaworthy and those that aren't really ready for sailing yet. But what happens when the tide goes back down?

## *A new problem*

There's a ton of work, and so we don't have to worry about the hacks that much – there's more than enough work to go around for everyone. However, the bubble we're riding is going to burst in a few years, and prices are going to drop as supply and demand attempt to synchronize. We'll be back in the same situation we were in not too long ago. And, even worse, we're going to be facing competition from all the hacks that were left on the beach at high tide – "Oh, he's charging you $X? I'll charge you ten percent less, and you don't have to pay mileage, either!"

And so we'll be back in the same situation, with a slightly different cast of characters: How do we demonstrate significant added value? How do we differentiate ourselves from the hacks? The answer is the same as what I was going to propose in my follow-up to the "The Starving Artist Club." In the previous scenario, we had a 26 year old whose primary responsibilities are (1) adding numbers in a spreadsheet, and (2) following the Packers during their quest for a repeat Super Bowl ring, and whose company is charging him out at $185/hour. And we had a 44 year old developer whose been delivering dBASE, FoxPro, and Visual Studio solutions for a decade and a half, is tightly integrated into the processes and mechanisms of the companies she works with, and whose company still pressed to justify a billing rate of $100/hour.

## *The answer*

The answer is to change the customer's perception of you. Make them see you as a professional delivering software development services, not an overgrown Lotus 1-2-3 expert.

Are you a technician? You know – one of those folks with a computer science degree but who hasn't read a magazine or a book in years? Or a professional? Someone who treats their work with the same care that other professionals – doctors, lawyers, accountants, architects, engineers – do.

Do you understand that you have to continually invest, not only in the specific tools of the trade, but in the practices surrounding the technical expertise? And, more importantly, do you communicate that to your customers?

Do your customers see you as a cowboy, slinging code and shooting the bad guys in a quick, but haphazard fashion, or do they see you apply proven techniques and documented past experience to produce a repeatable result? I know, that's starting to sound like the SEI eggheads have been pumping their white papers into me. Let's look at a real world example.

## *The audit as example*

For example, a potential customer often asks us to audit an existing application. This situation is fraught with peril, as you have to walk a fine line between honest criticism and political astuteness. We've all seen the application where we get to the third or fourth screen, and either starting laughing out loud, or just shake our heads in pity.

Only to find, of course, that the guy showing you the application was the original author, and is pretty damn proud of his work, since, after all, he did it in his spare time, armed only with the reference manual and an aging 286 at home. Or that perhaps the programmer was the boss's favorite nephew who has since gone onto better things, like college. (Isn't it truly frightening to see how often this happens?)

We go into audits with a preprinted checklist of things to look at, before we've ever seen the application. This forces two wonderful results. First, it provides the customer with a level of confidence that we've done this before, and second, it helps us produce a repeatable result – we tend not to leave the customer's site having forgotten things. You know – when you get to the parking lot and then slap yourself upside the head, going, "Damn! I forgot to ask them if they have any local printers!"

This method of using a formalized, documented process to produce higher quality results is something you're likely already doing here and there. Using a naming convention? Using an application framework? Have a standard procedure for interviewing employees? There you go. But don't assume your customer knows it – they don't.

## *The need for formalizing and documenting your processes*

You can apply this same concept to every step of your business. Do you have a process for answering the phone to handle inquiries? When you make a sales call with a potential customer? When you deliver a prototype? For your testing staff once they're finished with a module?

And when you document this process to show your customer how you do it, you are helping them understand that they are buying not just another clever computer programmer who has memorized the reference manual and can type 134 keywords a minute, but a business person who can deliver a complete product. And as much as this sounds like marketing talk, you're delivering solutions that are independent of your expertise with a specific dialect or language. Your value to your customers transcends the ability to use a specific widget – your process doesn't change simply because you're using VFP 5.0 instead of VFP 3.0, or Visual InterDev instead of handcrafting HTML.

(By the way, while this sounds like I'm just targeting the independent developer with this discussion, I'm not really. Corporate developers have the same set of issues – too much work, too many 'customers', and competition from hacks (i.e. the PC "guru" that seems to inhabit every department). You simply don't send out invoices. By differentiating yourself from those hacks, as well as perhaps your peers at your company, you can place yourself in the driver's seat. This means a number of types of rewards – such as money and perks, as well as less tangible items, such as bigger and better projects.)

## *I don't have any time!*

But back to the issue of formalizing and documenting your processes. This is a lot of.

But we're faced with a very opportune time to do this, because we can be making a healthy living at the same time. Then, when things get tight in a couple of years, those bottom-feeders who came in during a boom cycle are going to starve.

Yeah, yeah, yeah, I know. "When am I going to find the time to do this?" This is simply a new version on the age-old question of finding time to market yourself. If you neglect marketing while you're riding a fat, juicy contract, at some point, you'll find yourself at the end of the job with no work in sight. So you have to allocate a bit of time each week to market, no matter how busy you are. Same thing here – it may seem you have no time to do this – particularly now, since you're so incredibly busy.

But you simply have no alternative. If you can't find the time to formalize and document your processes, run, don't walk, to the exit of this industry. You have to change the customer's perception, and you have to do it now.

To sum up, the answer is to change perceptions by demonstrating the extra value you're adding. Yes, it's a lot of work and it's time consuming. So let's take advantage of high margins available now to do this extra work while we "can afford it," and thus be well positioned a few years down the road.