

# The Starving Artist Club

Whil Hentzen

Three blocks to the west of me, there lives a fellow in his early 20's who graduated from a local college with a low B average. He works for one of the national accounting firms, and spends his day adding columns of numbers in Excel for clients and talking about football over the cubicle wall. Nothing special, no specific genius, no remarkable attributes that separates him from the other tens of thousands of bright, ambitious kids who are shooting for perhaps a partnership or, failing that, a staff position at a client firm. And he bills out at \$210 per hour.

We spend our days and nights writing systems that can have a major impact on the way our customers run their business. However, (with the exception of a few major metropolitan areas), a skilled Visual FoxPro developer with ten or fifteen years of experience is hard pressed to approach triple digits. But an individual with little experience and no specific distinguishing traits routinely bills out at \$150 to \$250 an hour - rates that are virtually unheard of in our business.

The incongruity is appalling.

Why is this happening? And, more importantly, what can we do about it?

The *why* is fairly easy to explain, albeit a bit depressing. First of all, the barrier to entry in our business - software development - is low: \$1500 for a PC, \$99 for a copy of FoxPro or VB, and \$20 for business cards. And enough chutzpah to claim expertise in a profession they may not have known existed a week before. Actually, you can go even lower - I once ran into a "developer" who didn't have business cards - "I run a pretty lean ship and that's an expense I can't afford - I can't just spend money left and right."

The other professions - law, accounting, medicine, mortuary science - have higher barriers - including required schooling and degrees, exams that must be passed, and ongoing educational requirements. These barriers don't exist in our profession, and, as a result, there's always the new kid on the block willing and able to undercut rates - "He's charging you \$40 an hour? I'll only charge you \$30."

I know of a developer who started out at \$60 an hour. He wanted to charge \$75, but since he had just become a consultant (he had spent ten years in the corporate world), he thought that he might not be able to get that much. (Notice the lack of concern of whether or not potential customers would be getting any value at either of those rates.) I was having lunch with him one day, and he talked a good game for a bit, until I mentioned the term "normalization." A blank look came over his face. He'd never heard the term before.

It was all I could do to not lose lunch right in front of him. He simply had no concept that customers should actually be getting *expertise* from a consultant. And as you know just as well, this fellow is not an anomaly.

The second reason is a low level of sophistication on the part of our customers. I'm not saying they're stupid or naïve - they're just new to buying custom software. In the olden days, about 15 years ago, the only people who bought software were the high priests in the glass cage - and that was their business, so they were reasonably proficient at it.

Then the PC came out, and everyone needed a hand, and contracted with anyone else who had read further in the manual. In a few years, everyone had a nephew "who knew about computers" and could help out with a program one of the departments needed.

After a decade of buying spreadsheet templates and single user mailing list databases, users started to think they understood what custom software development was about. After all, they even wrote a program or two themselves (that worked! (most of the time)) - how hard could it be?

But there's a wide gulf between a single user inventory tracking program and a complex, professionally designed and written application, and most users don't have the experience to tell the difference. A shoddy application one step away from disaster can appear to be, on the surface, the same as a rigorously designed, built and tested application. Why would anyone pay \$160/hour for something that looks pretty much the same as something built by an individual charging \$25/hour?

There's a company in Milwaukee that was going to have a clerk design a system that would track all of the lab results in their QA department. This clerk had just finished an assignment, and so, to prepare him, they sent him to one of those 10-week community college classes to learn all about FoxPro. Once finished

with 30 hours of training, this person would be perfectly qualified to design, code, test, and implement a system that would be relied upon by the whole department, right?

The third reason is that there are no uniform methodologies. Other professions have, for instance, a code of ethics and a generally accepted set of standard practices. But our business, partly due to the rapid pace of change, doesn't. As a result, a developer can always cut corners (either knowingly or unknowingly), and it's difficult for the customer to tell what they're getting. Thus, they're reluctant to pay what they perceive as "top dollar" when the end result may be gold-plated instead of solid gold.

I liken our position to that of a talented, but under-appreciated artist. One of those undiscovered geniuses hawking their wares on Lafayette Square in New Orleans. We obviously have a job to do if we're going to move out of the six story walk-up co-occupied by cockroaches and a forty year old coat of paint. Next month – the plan!