

A Guide to Being a Consultant

Jack G. Ganssle
jack@ganssle.com

The Ganssle Group
PO Box 38346
Baltimore, MD 21231

How do you make money in the embedded systems world?

I hear this question posed constantly. Friends who own tool companies often wonder if the market is large enough to make the battles of business worthwhile. Engineers (those actually *doing* the work) complain about career stagnation and limited salary growth. Want ads offer few open positions, while my desk is swamped with an unceasing flood of unsolicited resumes, each with its own sad tale of angst hidden between desperately optimistic lines.

Let's face it - our business is so specialized that most non-technical people have no idea it exists. I've had to develop a boiled down story, simplified and almost meaningless, to tell the great unwashed of what I do for a living. It's just not worth the trouble to explain about embedded processors, when to a dentist or lawyer the word "computer" conjures only the image of a PC.

And yet... the products we embedded folks design underlie the fabric of civilization. It seems every electronic product has an embedded micro that somehow creates direct benefits for hoards of consumers. Information processing, sometimes quietly buried away in an innocuous kitchen appliance, is the great paradigm of the late 20th century. It's almost like the start of the industrial revolution, where those who built the cars and made the steel slaved away for a pittance, while others reaped the benefit. Sure, hardware and software engineering yields far more than slave wages, but is the pay really reflective of the value of our contributions to society?

And so a great number of engineers find their way to consulting to fulfill their immediate needs for a job and to chase those big bucks. Consulting's typical \$100 per hour (or so) rates look so appealing (uh, \$100 times 40 hours per week times 52 weeks... hey - I'll be rich!) that few can resist the lure.

In 1980 a friend and I quit our engineering jobs, and in a grand, expensive, and foolish leap we rented office space, hired employees, and became... consultants! Our only wise decision was to do this before either of us had children, because we starved.

The rented space was needed, we rationalized, because our clients would need to see a respectable company before forking over their cash. We spent endless hours building walls in the unfinished R&D space. I wonder if those walls are still there, and if they are still full of the empty beer cans we used for, ahem, insulation. In fact, in the years we were in business few customers ever saw the facility. Every month, though, the landlord arrived with his hand out, depleting our meager resources.

For the first weeks we stared at each other. What do we do now? How do we get customers? Who is going to pay all of these bills? We ran ads. No one responded. The yellow pages produced no inquiries. Small contracts from our former employers kept us miserably operating, until time taught us to sell, sell, sell. We started banging on doors,

making cold calls and getting no end of rejection. A little work started to trickle in. More networking and working friends of friends of friends brought more in.

As the work backed up we learned the next lesson - fixed price contracts are a horrible mistake. We did an audiometer for \$3000, designing and building the hardware, and writing all of the code. A grossly underbid \$12k steel mill job kept the paychecks at bay for months. To make any sort of decent living on fixed price work one has to bid high enough to cover surprises, and one's estimation ability must be prescient.

As time went on we honed the art of estimating software and hardware to a fine skill. Unfortunately, few private companies cared to enter into multi-hundred thousand dollar contracts for a product's development. Most seemed happier paying us hourly, even though the total billing would be about the same as for a fixed price contract. Humans are a perverse and puzzling lot - but in business, you must deal with the customer's expectations, needs, and oddities. The customer, is, after all, king.

Only the government seemed willing to foot large fixed price efforts, but we found it difficult to get those fiercely competitive contracts. A formula we hit on accidentally worked like a charm: write proposals, on spec, as a subcontractor to a large operation. Let the big partner handle the peculiarities of government work. After the award of the contract we'd get our share of engineering work.

After my partner and I went our separate ways I continued consulting, alone, and put these painfully learned lessons to work. I worked out of the house, keeping overhead low. My company had no employees other than myself. I secured a good answering service to create a weak but valuable business-like phone service. All of my billings were hourly, and most of my business came as a sub to large corporations involved in government contracting. The money was good (finally!), but I hated the work.

Fun and Money

There are only two reasons to work for a living, and I believe both are necessary conditions: fun and money. Who wants to be a slave, working like a dog and hating every minute, just to put food on the table? The opposite is only marginally better - having a great time, but not having the means to survive.

Everyone has different fun and money needs. Donald Trump might need an annual 8 figure income to be happy, whereas others are satisfied in the low 5s. Success of a consulting business (and in life) starts with an introspective, soul searching analysis of your own psyche. What do you want? How do you get from where you are, to where you want to be?

Even the smallest consulting company needs a vision to march to, something you can use as a yardstick to plan and measure against. Though everyone loves to hate Microsoft, I have enormous respect for Bill Gates. He defined a vision and spent years pursuing it. His dream of a GUI yielded the abysmal Windows 1.0, followed by the equally useless

2.0. But Gates kept at it, following his dream, and eventually producing versions that have quite simply taken the world by storm. A lesser man might have given up after the first or second debacle.

A reasonable consulting vision might be to generate a reliable \$90k income. Another might be to grow the firm to lawyer-like status, with the founder just living off the fruits of the labors of hired guns. Either way, the vision must be specific, so you can develop a reasonable plan to attain it. You'll surely guess wrong about many things, but the right goal lets you periodically stop, assess progress, and implement new plans as needed.

Far too many consultants have no firm goal, and thus drift along, never achieving much success. Know where you are going, develop a plan to get there, and implement it. Regularly monitor your progress and take action to get back on course, for even the best planners make massive mistakes.

Sales and Competition

You will go out of business if you are not constantly selling. My partner and I repeatedly made the classic consulting error of somehow, through heroic efforts, obtaining a contract, and then simply doing the work. The job was always complex enough, somehow, to require both of our efforts for its duration. We'd submit the final product and invoice... and then starve until another contract came around.

The vast majority of consultants go through this boom and bust cycle. Everyone complains they cannot sell while working on a job, so blindly work on, knowing a financial cliff looms even in the best of times.

Selling is as important as switching on the computers in the morning. The excuse "but I don't have the time" is simply unacceptable. As the only way to avoid those horrible bust times is to sell constantly, you *must* aggressively sell or market your services every single day. You have no choice, so plan to run the business with sales as the top priority. If time is precious (as it always is) a solution might be to bill fewer hours each day, leaving time free for sales. Rest assured that those "lost" hours will be recouped in lack of downtime.

Never forget that consultants live in a fish tank of vicious competition. Every unemployed engineer has a de facto shingle hung out. Many fully employed designers consult on the side, often for ridiculously low wages. It's easy to find a consultant; it's not so easy to insure that you are the one that is found.

Scott and I made a fundamental mistake in forming our partnership. Both of us were techies, a fact that in our naivete we thought would make us more productive. Wiser heads would have taken a third partner on board, one devoted entirely to marketing and sales. This third person could have kept our in-baskets full, thus insuring ongoing strong receivables.

But wait! you say. If I sell all of the time, what happens if I'm too successful? I can only work on one or two jobs at a time - suppose a dozen come through?

My feeling is it's easier to turn work away than pay the bills those times there is no income. Besides, success breeds more success. If the word gets around that you are in demand, then more will want your services. You might even become a scarce commodity. Raise your rates!

You can deal with success in more effective ways, of course. Hiring engineers is an alternative. This creates more challenges, as someone (i.e., you) must manage their work, burning up yet more of your limited time. Some outfits very cleverly develop networks of consultants they can draw on in peak periods, raking a percentage off the top. If you serve as prime contractor for these outsiders, you'll be held responsible for their work, so select them carefully.

How does one sell consulting services? This is a tough nut to crack. Some companies advertise in national publications, an option generally too expensive for one and two man shops. Others beat the want ads to death, using these as a source of intelligence into business' needs. I doubt that any pat answer exists, but it seems that the best companies thrive by developing and exploiting relationships.

Get around. Meet local business people. Join an organization that might have potential customers involved. A natural is the local High Tech group. Here on the East Coast it seems every county has some sort of High Tech Council. Some are better than others - research them and spend your time with the one most likely to produce valuable contacts.

Remember that your goal in networking is to meet people. Never join and just scan the newsletter. Get involved. Chair a committee.

Participate in causes. Did you know that Gates joined the United Way board simply to meet John Ackers of IBM? What works for the big guys can work for us as well.

Call on engineering managers to discuss their problems. Count on rejection. A lot of rejection. So much you'll wonder why you do this. Remember that rejection is never personal, and that one win in a hundred calls could keep you in bacon for years.

Do mailings. Follow-up on them. Contact the State for lists of technology companies. Some counties maintain decent lists as well.

Yeah, all of this takes a lot of time. That's why I believe in taking on a partner dedicated to sales. No one has enough time to sell and do the work. Know what you do well, and do it. Surround yourself with others who complement your skills.

Life becomes sweet when that 2 year 40 hour per week job finally comes though. Nurture the customer. More work will certainly follow if you treat them well. Do a good job, and make sure they know just how good your work is.

Beware, though, of the big customer. Don't allow your business to become dependent on one company for most of your income. New management could decide to fire all outsiders (it happens all of the time). They might go broke, possibly even leaving you stuck with unpaid bills. Big customers are important and wonderful and awful, so develop other clients as well. Never place all of your bets on one role of the dice.

It seems the very best consultants, those who are tremendously honest, whose customers are satisfied, have little trouble getting new business. There's no more powerful marketing medium than word of mouth advertising, something you can improve by asking your happy clients for referrals or referral letters.

Referrals are especially powerful in the embedded world as the service we provide is so nebulous. After writing tens of thousands of lines of firmware or designing an FPGA for a customer you've mastered many technical and business issues, yet how can you convince a new prospect that you're good at this sort of thing? Every engineer claims expertise; far too many are unable to deliver working goodies on time and on budget. A referral, from someone the prospect trusts, is the most powerful sales tool you'll ever find.

I could ramble on for chapters about other small business killers like poor cashflow management, but these are all secondary to sales. Remember one thing: success at consulting comes from never ending selling. Do this and you can generate a steady and appealing income. Forget it and things may be rosy for a while, but the ceiling will certainly fall in.

Customers

Without a customer you have no business. Though many people become consultants by circumstance (read: getting fired or laid off), when you make a conscious decision to change your life and strike out solo find a customer *before* cutting lose from a paying job. Without a customer your fledgling business is bleeding, in the process of dying.

Customers are tough to find. You're competing against an army of others, some willing to cut rates to suicidal levels just to get a few bucks in the door. It's important to differentiate your services from the rest of the pack.

It's best to focus on an industry. Too many engineers hang their hats out as generic "embedded systems designers for hire". Most of your customers simply won't care about cranking schematics or code, and many more fear the pain of training an outsider in the intricacies of their particular business area. And we've got to recognize a painful truth about our embedded careers: few outsiders have a clue what we do and why... and fewer care. It's hard to get a product manager whose embedded knowledge approaches zilch to care about our expertise with FPGAs or Forth.

The jack-of-all trades simply looks like an idiot when tossed into a new environment. How can you possibly hope to sell your services when you can't even speak the customer's lingo? Make promises about meeting FDA software standards when you have no idea what these entail and you're courting bankruptcy. If you're talking to a paper mill operator and don't read the industry trade publications you'll be forever in the dark about their needs and pains.

Instead, pick a sharply-defined arena where you have clear expertise. It's hard to sell yourself as a C code guru, but if your forte is writing firmware for medical instrumentation, then you'll have a proven track record in an area where your customers have needs that you can immediately fulfill. If you're and expert at spacecraft navigation software you'll know precisely which companies could use your services. This vertical market focus is like rifle-shooting, as the universe of potential clients is clear and easy to target.

A more horizontal yet equally effective idea is selling yourself as *the* guru at a particular type of technology. Companies are hungry for TCP people now that marketing hacks seem to want Internet connectivity in every embedded product. A shingle that reads "TCP/IP implementations on sale here", backed up by a list of successes, will attract customers like moths to a light.

Sell your unique proficiencies, and always continue to hone that expertise to an even better edge.

Being Businesslike

There is nothing more critical to success than selling yourself and your business. Unfortunately, in business *everything* is important. Without sales you cannot even get started. Without engineering you cannot deliver. Without proper cash management you'll choke on success or failure. Even little things are important, like how your phone is answered.

I once found a full page color ad in a major PC magazine promoting Bell Atlantic's "Airbridge" service, a way to connect your laptop to a cellular phone to get access to E-mail and other on-line services. It seemed like a neat idea so I called the 800 number and was presented with a deep menu of electronic messaging options. After spending too much time experimenting with different selections a fax listing available Airbridge documents arrived. Another call to the 800 number, more navigation of the menus, and I entered the appropriate document numbers. Surprise! Each entry generated an "Illegal document number" response.

The ad had neither address nor non-800 number listed, so I gave up in frustration. A few weeks later Bell Atlantic had yet another ad, quoting a customer soooo happy with their service that he wanted prospective customers to call to hear him sing the Company's praises. Just for yuks I called on my Bell Atlantic cellular phone... only to get a "I'm sorry, this number is not in service" message.

The moral of the story is to be accessible. Make it easy for people to contact you. Most won't try this hard! How often have you called a business to get an answering machine, confusing voice mail, or a sullen receptionist? This immediate turn-off is a guaranteed way to lose business fast.

The most successful consultants I know either have a full time receptionist or a wonderfully friendly phone system that gives their clients a fast and easy way to get in touch. One option is well planned voice messaging that you check hourly. Another is a cellular phone that you *always* answer promptly. Avoid answering machines. Beware of having a spouse pick up the phone, unless he is trained to answer in a businesslike fashion identifying the name of your company.

Crying kids or dogs barking in the background immediately imply "small time". A quiet private office is crucial. Success requires creating an image of sophistication and competence, even if this is a part-time business that you started yesterday. Long ago, while I was part of the love and peace generation, we eschewed "image", thinking that it was a sort of sham perpetrated by "those losers in Madison Avenue". We were wrong. Image is a part of human psyche that *everyone* must deal with in some manner. I blush to think of our denunciations of "image" while we oh-so-carefully cultivated long hair, beads and tie-dye shirts.

Image in business is no more than making your customer comfortable about dealing with you. This might mean wearing a suit to meetings with suited executives; creating a phone system that reeks of "professional", and generally acting in a quietly competent manner, even when you haven't a clue what is going on.

Is this a fraud? I think not. No incompetent or even marginally able person should hang out a consulting shingle. To do so is fraud. However, if you really know what you are doing, and have a reputation of being able to make things work, then sometimes it makes sense to act capable while admitting to being short a few facts, in the expectation that you'll soon figure it all out and give your customer value for their money.

And, if it doesn't work out, simply do not issue a bill. Above all, going into business yourself is a commitment to take risks, one of which is exemplified by the "no cure, no pay" concept used by salvagers at sea. Yes, some contracts are hourly where you'll be paid just for showing up. In many cases a potential client will contract with you to fix a short term problem. It seems unethical to charge for failure.

And, it's great marketing to be able to tell the prospect "hey, if I can't help, just don't pay me". The risk is all yours, a very appealing thing to any customer.

Another aspect to image is creating literature that promotes your product or service. Too many consultants rely on a crummy Xeroxed resume. There are a lot of folks beating on doors looking for consulting work - your name and company must somehow stand out from the others. Invest in a decent set of brochures that gets your message across.

Money

Cash is the grease of business. You'll need far more of it than you can imagine. Before hanging out your shingle come up with a plan to get access to money in emergencies.

Forget venture capital - only a tiny number of firms ever get any; most of these are run by business geniuses with armies of contacts, lots of previous experience, and a killer product. The only sort of VC that is possible for most of us is from family and friends.

The second avenue is a bank. Banks are happy to loan to small businesses - as long as you have sufficient collateral. Too many new entrepreneurs moan about collateral requirements, but this is simply a fact of life in the banking world.

For a small business the bank will want you to cosign a line of credit. Then they get your house and other assets if the company folds. Further, you'll need an asset that is very liquid - accounts receivable are probably the most frequently-used form of collateral, though sometimes you may have to sign over a personal T-bill or other asset as well. But it's very hard to get a bank to fund a tiny business on receivables. They're more interested in your house or investments as collateral.

I've found that smaller, local banks are easier to deal with than their mega-cousins. No banker likes to deal with an unknown, so make sure that you are visible. Get to know your target banker by getting involved in community activities and by sending them information about your business. Take a banker to lunch once in a while.

Remember, though, that bankers hate new businesses. Figure on starting off with no bank backing, but plan on generating sufficient profits that in a year or so you can go back to the bank with a history that you are proud of. If you can show cashflow and collateral, getting a line of credit will not be a problem.

Bankers really do want to make loans, but work within strict regulations that limit their ability to take risk. A successful business will need a line of credit at some point, so start your business with a plan to cultivate a banking relationship. Above all, start before you really need the money. Plan ahead, since in desperate times no one will answer your calls.

I think most small businesses are initially financed by credit cards. These will be personal cards, as no organization will give your fledgling outfit significant credit. Some businesses get *lots* of credit cards, and pay the yearly fee with a smile, realizing that the plastic might keep them alive someday. It's not unusual to talk to a business owner who has \$50-100k in available credit on their credit cards.

But debt can kill a business quickly. Better: don't spend. Every time you feel the need to buy something or commit to some expenditure, ask yourself if this will help you make more money.

Don't buy stuff. If you're not a Midas be a miser.

Keep your overhead low. It's oh so tempting to get that nice Dell server for only \$200/month and that leased fax machine and a swell oscilloscope on credit - but by the time the loans are paid the equipment may be obsolete and you're stuck bleeding money forever. When the first big check comes in it all too often feels like party time, but resist normal spending urges and use the money wisely. Dry spells are a comin', and without some savings you'll die.

Cash management means knowing which bills must be paid on-time. Never, ever, miss any IRS payment as the interest and penalties swell the amount due to dizzying levels. It means paying the employees first - even that *&%& engineer who is so late delivering working code that the customer calls you at home every night about it - and paying yourself last.

Cash management means keep a spreadsheet of all expected income - and *when* you *realistically* anticipate receiving each check - as well as all bills. Develop a budget, a model of how the business operates that you tune from month to month so you know, with great accuracy, how much cash you'll need for the next 90 to 120 days.

Know what's coming in and what's going out with precision. If there's not enough money to cover all of the bills you'll be able to use this information to juggle the numbers, and to decide what to pay when.

Cash management means maintaining your accounting files religiously. Money is a big part of the reason for starting a company, so treat money matters seriously. Even a one-person shop should use a decent program like QuickBooks to track all income and expenses. It's tedious, but essential.

If you use an outside bookkeeper, or as the business grows and you delegate this role, remember that honest mistakes will happen, and that there's some chance of your being a victim of someone's dishonest errors. Either can destroy all of your hopes and dreams. One company owned by a friend found \$57k of incoming checks diverted into the bookkeeper's personal account, \$57k that was never repaid. Delegate accounting, but develop your own quick cross-checks to insure that the numbers make sense.

Employees

At some point many consultants become overwhelmed with either work or the tedium of office maintenance and hire employees. Be wary! Payday rolls around every week, even those weeks when the bank account is squeezed dry. Few employees tolerate even a day's delay in getting their check. Somehow you will *have* to come up with their pay each and every week. Your own check will be a distant second in priority.

One friend tells how for 68 straight weeks the night before payday his company had no money, yet somehow enough showed up just hours before his nearly 100 employees

started cashing checks. Somehow he survived, though even a one week glitch would have made the whole company walk.

Poor cashflow kills more small companies than anything else, and the biggest impact on cashflow is payroll. Before hiring, be sure you can make payroll week after week, year after year. If cashflow is good salt away the profits to use as a rainy day fund. Get a line of credit with the bank. Just be sure that you have a way to get a reasonable amount of emergency cash to deal with the hard times that will surely come.

Now that the government intrudes on each little facet of our lives you practically need a lawyer to wake up in the morning. Your business may have to conform with hundreds of laws regulating relations with employees and the public. Worse, sleezeballs will try to take advantage of each opportunity.

Most employers I know have been involved in frivolous discrimination or harassment actions (I could write pages, but will spare you, gentle reader). Generally the government pays for the prosecution of the "case" brought against a business while you have to pay your own way - even when you win. Since there really is no defense against such abuses, I recommend that you treat your employees as fairly as possible, using the Golden Rule as a good behavior model. Then, hire a good lawyer and build a war chest for the inevitable battles.

One company I know is really a shell organization that owns no assets. They contract to other businesses all owned by the same person. The idea is that the shell can fold in a heartbeat if it loses a big lawsuit. Though perhaps this is a cynical view 21st century America, there may be some merit to the idea.

Employees do bring a new kind of slavery and of freedom to your life. Slavery, because you must provide them with a kind environment and the resources they need to be productive. As boss, you'll find yourself spending more and more time managing and less time selling or working on billable projects. Freedom, since you can delegate many routine tasks and concentrate on the more creative (and enjoyable) problems you like.

Hire carefully, and then treat your employees well. Give them a stake in the success of the organization via bonuses or stock. Fire losers early before they disrupt moral, and nurture the winners.

Planning

In business, as in life, it's less important to *do things right* than to *do the right things*. How often have you seen a consultant devote weeks or months to satisfying someone paying only a few thousand when the rest of the business goes down the tubes from neglect? This isn't advice to screw your customers; rather, it's a mandate to spend your limited energy on what's really important.

And yet, it *is* OK to fire a customer. When 50% of your time goes to a project producing 10% of the income, think! Is this the wisest way to move your business ahead? You are entitled to make mistakes; what separates the men from the boys is the action you take to correct those errors.

It's impossible to do the right things unless you know what those things are. Create a plan. A detailed, written plan. Even if your goals are modest, perhaps to simply consult to one particular customer, without a written plan you're like a ship without a compass, steaming along full speed ahead, but with no clear course.

Technical people hate to draw up business plans. It's the same defect that prompts us to jump in and start coding before completely designing a system; indeed the two symptoms are the same problem in different guises. Planning, or designing, is less stimulating, with fewer immediate rewards, than jumping in and doing.

In one sense a company is harder to design than firmware. The code will (hopefully!) at some point be complete. The goal, as represented by a specification, is reasonably clear and objective. A business is never complete; it's a *process* that goes on and on. Like Miss America contestants many CEOs now proclaim noble goals for their operations ("I want world peace, a nice environment for employees, and a fitness center for the executives"); the reality is that the process implemented by any business operates to satisfy some sort of well defined goals.

Publicly held companies generally must maximize shareholder values, as measured by the dividend or stock price. Small outfits, like one operated by you as a consultant, will have other goals. Making a living is a minimal objective. Making a fortune is something more exciting, tangible and easier to plan for, though harder to accomplish.

Business planning starts with identifying your personal goal. Pick something ambitious, something that you're willing to dedicate a big chunk of your life to. Make it something concrete, like "making \$150k a year and taking 2 months vacation."

Know where you want to go. Then, and only then, develop a plan to get you there.

Shelves in libraries groan under the weight of books about developing business plans. A dozen computer companies happily sell software that claim to do the work for you. The books have some value as idea generators, but there's no book or program that will significantly ease the trouble of generating a decent plan.

The books tell you to start with an Executive Summary, to move on to a description of the products or services you'll produce, to lay out a careful analysis of the competition, etc.

My advice is to ignore the books. Who needs wonderful prose? Why waste weeks agonizing over writing styles? A plan should be something that's quick to use, that lists what gets done when and by whom, and that's dynamic -- it will be shaped and reshaped

by forces in the marketplace and your own evolving understanding of the requirements for success.

A plan is a simple thing. List your goals -- specifically, clearly, and with a time frame (“retire at age 40 by selling the business for \$yyy”). Figure out the major objectives needed to achieve these (“sell \$xxx of products or services in years 1, 2, 3,” or “increase the company’s net worth to \$zzz”). Invent a method to get there (this is the hard part!). Identify all of the major action items needed, with due dates, to reach each objective.

A typical business plan lists your dream first. Then it identifies major goals needed to reach this dream. For each goal, list the objectives you must satisfy to reach that goal. This might be to generate 100 leads per month from magazine ads, and then using certain sales techniques (outlined in the plan) to convert 5% of these leads to sales.

The plan must never make random assumptions. How will you get those 100 leads from advertising? Call various magazines pertinent to your industry and pry for information. How many leads does that half-page black and white ad from competitor A do? A little spying will give you a pretty good yardstick for accessing lead flow versus ad size - and thus also give you a rough budget for that part of the plan.

The plan leaves nothing to chance and questions every assumption. Giddy sales projections are backed up with solid assessments of how they'll be achieved. Dates are assigned to each action item so you know *what* must be done *when*.

The plan is a waste of effort, though, if you don't use it regularly. At the very least refer to it monthly to extract that month's goals. Pour these into your Daytimer and make sure each one is accomplished on-time. Does the plan indicate that a press release is due this month in order to increase lead flow? That's an action item!

No plan will stand the test of time totally intact. Business conditions change. New competitors arise. Often the plan itself is simply flawed in some way: perhaps you learned something new about the industry, or perhaps you have a new brilliant insight into achieving an important goal. A useful business plan is a living document. Amend it constantly based on what you learn. Like a knife, it must be continuously sharpened to be effective. I've learned to keep a copy with me in my portable Daytimer so I can study and amend it while out of the office – when the best ideas occur.

Above all, be realistic! Plan on failure, since some strategies will simply not work. If the plan is constructed like a house of cards that rely on each card being in place, then the entire plan will crumble when a tiny obstacle arises. Be flexible.

Planning, though, is the easy part. Most of us fall down on execution of those plans. Unless you can make unbreakable commitments to yourself to achieve what you've set out to do, simply forget the whole idea of starting a consulting practice (or any other type of business, for that matter).

When I was in the development tools business we actually maintained two different plans: the real one, and a beautiful prose-ridden tome written just for the bank. One a year we had a mad scramble to update the fancy one, recognizing that unless we deal with the bank on their terms they won't deal with us. Part of the cost of doing business is doing silly things other important allies expect.

Even the best business blueprints suffer from our own narrow perceptions. Get out of the office from time to time, well away from phones, email, and fax machines, and spend a day rethinking every aspect of your strategy. Start with the most important: is your grand goal still where you want to end up? It's OK to be wrong; just realize your mistakes, if any, adjust the scheme as needed, and move on.

Create Value

Technology companies have some unique opportunities to build something of value in new and different ways. A consulting business, for example, is engaged in the process of developing new products for hire. Anything that helps you do this more efficiently is worth serious money, and ultimately increases the value of your business.

The mistake most consultants make is to work steadily for their customers. It's insane, really -- consultants bill themselves as experts, as the magic that will solve all sorts of problems for their clients. Yet these same experts rarely devote even a microsecond to helping their own businesses.

You can get a jump ahead by devoting time and resources to tuning your engineering system. Constantly invent new ways to work faster; each minute saved translates into more profit and lower costs for your customer.

For example, never develop throw-away code. Consider the reuse possibilities of every routine you write. Resell this code shamelessly and profitably. When designing hardware use components you understand intimately. Otherwise, when each job needs a totally new UART you'll have to waste time learning yet another 100 page data sheet.

Don't take every job that comes along. Be selective. Work only where will increase your core competence, or give you something more than money -- important new class libraries, critical new contacts, or a knowledge of important new technologies. If someone will pay you to learn C++, go for it! Avoid dead end jobs like an embedded 8080 FORTRAN project.

Look for those that live on the cutting edge, or that will build your organization. Even a single-person company needs growth -- mental and spiritual at least. Proactively pursue it.

Honesty

Unfortunately, some consultants thrive on disaster. When clients are in pain, when their worlds are crashing down in rubble around them, consultants, rather like lawyers, prosper. The refrain “I don’t care what it costs, let’s find a solution” is the siren call to armies of slick briefcase-wielding folks from McKinsey and other big-name consulting outfits. We see similar issues from time to time in the embedded world, when a product just *has* to get done, no matter what. Perhaps the employees are too busy; maybe they are not competent, or their competence is in question from a series of continued schedule slippages.

All too often consultants respond with metaphorical rape and pillage. The smell of disaster brings out the greedy Scrooges in too many. Cupidity benefits no one; few of the situations I’ve seen of this nature ever result in a working product, satisfied customer, or a consultant with an intact reputation. Of course, like the barbarians of a millennia ago, many are content to scorch the earth and move on, content in their belief that the field of suckers is large enough for them to prosper.

We parents work hard to instill basic virtues - including honesty - into our young ones. Isn’t it reasonable to assume that most adults behave in a reasonably truthful and equitable manner? Sadly, from reading the paper and email from people around the world it’s clear that honor is a vanishing commodity.

I’m astonished to hear the press slam businessmen people; it seems all are capricious, determined to screw anyone to make a buck. The reality is that in business honor is the most basic asset one has. Without honor and trustworthiness business will cease. It’s amazing that we trust a PO, for example - it’s nothing more than a promise to pay, perhaps signed by a clerk on behalf of a higher-level person. Yes, there are lots of crooks in the business world, yet the vast majority of managers I’ve met are decent, honest people. Out to make some money, for sure! Determined to succeed, absolutely. But by and large these folks work from within the confines of a respectable moral code.

It’s important that we consultants behave similarly.

Embedded systems are tough. Problems will develop. Recognize that you cannot always meet customer expectations (the client wants perfection, now). Develop great communications skills to keep your client informed. Avoid the temptation to lie or to manage the truth a bit to keep that smile on his face; be willing to take the heat in the short term when you’re late or in trouble, knowing that respect will come only through the mutual struggle of bringing the product to fruition.

A friend called last week; he spent over 6 figures with a small consulting company whose charter was to develop a new product. Things ran late, bugs crept in, yet the message he always heard was “don’t worry, be happy, we’re on top of it.” As the scope of the disaster became ever clearer the consultants became more distant, hiding behind other projects in other cities, placing less skilled new hires on the work, while still chanting their “just a few more weeks” mantra. The product is now a year late and may never be delivered.

Though my friend surely could have managed these contractors differently, he was the victim of a small conspiracy of lies that spun an ever larger web.

The IEEE Code of Ethics is copyrighted so can't be reproduced here. Get it from their web site (www.ieee.org). But its third item is "to be honest and realistic in stating claims or estimates based on available data."

That's a pretty good plan for all of us.

Answers to Questions

These musings were published in various form in Embedded Systems Programming over the years. Readers generated plenty of questions, some of which I've answered here.

What do we do about aggressively stupid customers? The ones who think that tech support means free design consultation; that warranties cover improper hookup; or that devices with embedded controllers should be 'open source'.

Next to starving from a lack of business, this is possibly the most frustrating aspect of being an embedded consultant. Customers who aren't willing to pay the real cost of your services are welfare cases, not sustainable elements of your success.

I hear constantly from consultants who lose jobs because their price was 10 or 20% higher than a competitors'. Price shopping customers are our worst nightmare. Price is important only after all other aspects of a proposal are equal: a tender that excludes (or ignores) support, for example, is simply incomplete; the resulting price just does not cover real project expenses.

Some years ago a banker and I talked to a local group of owners of very small businesses about success and failure. We asked each participant why they had started their ventures. I'll never forget Bob the banker's exasperation that none answered the question correctly. *Businesses exist to make a profit.* Without a profit all companies fail. You're not doing this so the customer likes you.

Your inventory is your time; spend it freely and there will be no inventory left to sell. Customers must trade cash for depleting your inventory.

For this reason every consulting job needs a complete contract that covers all such eventualities. Spell out the "free" support level, and the hourly cost for more.

Some folks use a fixed price development contract followed by a mandatory retainer for support. The customer pays a monthly fixed fee for an agreed-on level of help. Exceed this level (perhaps to add product features) and it's time for a new contract.

The contract must clearly define deliverables. Is source code included? If so, do you retain the right to plagiarize at least some of the source for other customers' projects?

Who is responsible to maintain tools like compilers for future upgrades? Exactly how much documentation gets delivered? Though the penurious client might demand no docs, surely sometime in the future their needs will change. Better put an optional cost in for those documents up-front, so there's no surprises later. They may elect not to buy them today, but at least you've established their value.

Stick to your guns. There ain't no such thing as a free lunch; though customers may demand the world, ultimately they fail if you fail.

Is there a better way to manage my accounts receivable? As we've grown, we've had some horrific adventures in this area--ones for which my literal-mindedness left me poorly prepared. This week we dodged a serious bullet when a customer with a bad credit rating bounced a huge final payment check.

Customers with bad credit must pay before you start doing work. Credit problems indicate insolvency or an unwillingness to pay as required; either condition is intolerable to the small undercapitalized consultant. Your business exists to make money, to support you and your family. When you allow a customer to avoid payment you're taking food out of your kids' mouths.

Don't let fear of confrontation stop you from discussing these issues with potential clients. If no one else will extend credit, this is a clear sign that, no matter what promises may be made, the customer often will not, or cannot, pay the bills. Take action before you accept the job to insure that your needs (getting paid) are met just as surely as you'll meet the client's needs.

I believe that huge final payments are a sign that the consultant is putting too much reliance on a possibly capricious client. Most of us cannot afford to be the customer's bank!

Doing engineering work on a contract basis is like an uneasy dance between two hostile partners. The customer, rightly, really doesn't want to pay anything till the product is complete and demonstrably working. The consultant needs a constant flow of cash to finance what might be a months or years long project.

Progress payments can solve most of these problems, but only if you structure them in a way that clearly shows headway, in a sense the customer understands. Few care about how much code you've cranked; they want to see something that works. Structure payments around a rapid development process. Build systems incrementally, creating some product functionality early and enhancing it as the project matures. See "Rapid Development" by Steve McConnell, Microsoft Press, 1996 ISBN 1-55615-900-5 for ideas about building systems incrementally.

Some consultants require a retainer; the customer fronts some sum each month which the consultant works against. That's not a bad model; it has worked for lawyers for decades and no one complains.

My advisor tells me we need a better "image". He figures we should rent some nice space with fancy carpets and desks. Sounds expensive – what to do?

After firing your advisor, move back into your basement.

For most people, perception is reality. What they see – or think they see – shapes their opinions. But two inch thick carpet in a full service building is yesterday's model, not one any of us need today.

Most customers understand that home-based businesses save them money; it's their cash that pays for a fancy, unneeded, office complex. If they don't get it, show them.

Rent an office and you're paying an annuity that never stops. When business is bad – as will always happen from time to time, that nice office is doing nothing but bleeding your cash supply. You'll make the landlord smile, but it's wiser to keep your bank account fat and happy.

The cost of an office is far more than just floor space. Heating, lighting, telephones, extra computers (few of us give up our home machines so much of the capital equipment will be duplicated) and a host of other costs drain finances.

An outside office starts to make sense when employees enter the picture. It's hard to find room for a handful of people in most basements. However, some of the most successful entrepreneurs I know creatively expanded their basement offices to accommodate increased space needs. Their philosophy boils down to one of "should I put this rental money into my pocket or someone else's?" For a good treatise on how these sort of folks accumulate personal wealth, read "The Millionaire Next Door : The Surprising Secrets of America's Wealthy" by Thomas J. Stanley and William D. Danko, Pocket Books, 1998, ISBN: 0671015206.

Image *is* important, but the image clients respect most comes from non-stop professionalism. Deliver quality goods on time. Develop a stable of happy customers as references. An embedded consultant sells services that are quite ethereal and hard for most customers to understand. They'll buy from those with proven track records, those whose "image" is one of reliable fulfilling of promises.

Here in Kansas we've got a lot of engineers working as consultants. How can I differentiate myself?

I think too many engineers count on differentiation through technology. Few customers care if you're the Midwest's foremost PIC expert. Sometimes technology opportunities

seem to appear – with every toaster now getting an Internet connection perhaps there's a market for a TCP/IP whiz? Sometimes a limited window for an expert pandering his brain in this manner, but these windows close quickly and firmly. Now, for example, TCP/IP expertise might be replaced by any of the dozens of off-the-shelf libraries advertised in this magazine.

As a consultant who once would take a job – any job – for money, I learned that with each new product I had to learn a new industry. We did an audiometer, and slowly became experts in the behavior of the tympanic membrane. Losing money, in the process, of course. Instrumenting steel mills we learned all about the behavior of radiation in hot steel – at our expense. Building security systems we again picked up all of the nuances of what a customer expects in a typical alarm situation, while slowly losing our business to our poor management of these products.

Differentiate yourself by industry. Pick something you're good at – really good at – and specialize in that. Data communications, perhaps. Colorimetry. Plant automation.

Each of these, and all other fields, has its own lingo, its unique and important perspectives and requirements. Most have journals and magazines. Go into a customer's location without the specialized knowledge of the field and you'll look like a fool. Wise clients will recognize that you'll be on a learning curve, at their expense.

Conversely, visit a potential client armed with a deep understanding of their needs, plus proven success at similar systems, and they'll have an awfully hard time rejecting your proposal.

Several readers have written to tell how they are experts at, say, DSPs and want to sell this knowledge across a lot of different application areas. I'm inclined to think that it still makes sense to specialize in a particular industry, to apply this knowledge in just that one arena. Though technically there may be little reason to limit options, the hardest part of consulting is getting work. Focus your marketing efforts on one area. As the old sales saying goes, it's easier to get customers by rifle shooting (picking a few very specific targets) than by using a shotgun (trying to talk to everyone in every industry).

Better Firmware... *Faster!*

A One Day Seminar

April 23, 2008 - Chicago
Hilton, Oak Lawn
9333 Cicero Ave.
Oak Lawn, IL

April 25, 2008 - Denver
Embassy Suites Hotel
7525 East Hampden Ave.
Denver Tech Center
Denver, CO

May 19, 2008
London, UK
Jurys Inn Heathrow
Eastern Perimeter Rd.
Hatton Cross, Hounslow

Presented by **Jack Ganssle**, technical editor of *Embedded Systems Programming Magazine*, author of 6 books and over 600 articles

Registration form on last page of this brochure

Limited seating; sign up now and guarantee a spot.

The Ganssle Group
PO Box 38346
Baltimore, MD 21231
(410) 504-6660
fax: (647) 439-1454

register@ganssle.com
www.ganssle.com

For Engineers and Programmers

This seminar will teach you new ways to build higher quality products in half the time.

80% of all embedded systems are delivered late...

Sure, you can put in more hours. Be a hero. But *working harder is not a sustainable way to meet schedules*. We'll show you how to plug productivity leaks. How to manage creeping featurism. And ways to balance the conflicting forces of schedules, quality and functionality.

... yet it's not hard to double development productivity

Firmware is the most expensive thing in the universe, yet we do little to control its costs. Most teams deliver late, take the heat for missing the deadline, and start the next project having learned nothing from the last. Strangely, *experience* is not correlated with *fast*. But *knowledge* is, and we'll give you the information you need to build code more efficiently, gleaned from hundreds of embedded projects around the world.

Bugs are the #1 cause of late projects...

New code generally has *50 to 100 bugs* per thousand lines. Traditional debugging is the *slowest* way to find bugs. We'll teach you better techniques proven to be up to 20 times more efficient. And show simple tools that find the nightmarish real-time problems unique to embedded systems.

... followed by poor scheduling

Though capricious schedules assigned without regard for the workload are common, even developers who make an honest effort usually fail. We'll show you how to decompose a product into schedulable units, and how to use killer techniques like Wideband Delphi to create more accurate estimates.

Learn from the Industry's Guru

Spend a day with Jack Ganssle, well-known author of the most popular books on embedded systems, technical editor and columnist for *Embedded Systems Programming*, and designer of over 100 embedded products. You'll learn new ways to produce projects *fast* without sacrificing quality. This seminar is the only non-vendor training event that shows you *practical* solutions that you can implement *immediately*. We'll cover technical issues – like how to write embedded drivers and isolate performance problems – as well as practical process ideas, including how to manage your people and projects. *After taking this class you'll receive a certificate awarding you 0.7 Continuing Education Units.*

Seminar Leader



Jack Ganssle has written over 600 articles in *Embedded Systems Programming*, *EDN*, and other magazines. His five books, **The Art of Programming Embedded Systems**, **The Art of Developing Embedded Systems**, **The Embedded Systems Dictionary**, **The Firmware Handbook**, and **Embedded Systems, World Class Designs** are the industry's standard reference works

Jack lectures internationally at conferences and to businesses, and has been the keynote speaker at the Embedded Systems Conferences in both Boston and San Francisco. He founded three companies, including one of the largest embedded tool providers. His extensive product development experience forged his unique approach to building better firmware faster.

Jack has helped over 600 companies and thousands of developers improve their firmware and consistently deliver better products on-time and on-budget.

Course Outline

Languages

- C, C++ or Java?
- Code reuse—a myth? How can you benefit?
- Controlling stacks and heaps.

Structuring Embedded Systems

- *Manage* features... or miss the schedule!
- Using multiple CPUs.
- Five design schemes for faster development.

Overcoming Deadline Madness

- Negotiate realistic deadlines... or deliver late.
- Scheduling - the science versus the art.
- Overcoming the biggest productivity busters.

Stamp Out Bugs!

- Unhappy truths of ICEs, BDMs, and debuggers.
- *Managing* bugs to get good code fast.
- *Quick* code inspections that keep the schedule on-track.
- Cool ways to find hardware/software glitches.

Managing Real-Time Code

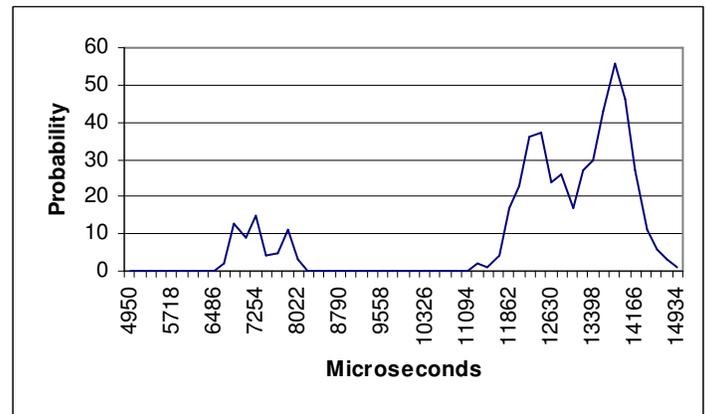
- Design *predictable* real-time code.
- Managing reentrancy
- Troubleshooting and eliminating *erratic crashes*.
- Build better interrupt handlers.

Interfacing to Hardware

- Understanding high-speed signal problems.
- Building peripheral drivers faster.
- Inexpensive performance analyzers

How to Learn from Failures... and Successes

- Embedded disasters, and *what we must learn*.
- Using postmortems to accelerate the product delivery.
- Seven step plan to firmware success.



Do your routines execute in a usec or a week? This function is all over the map, from 6 to 15 msec. You'll learn to write real-time code proactively, finding timing issues early.

Why Take This Course?

Frustrated with schedule slippages? Bugs driving you batty? Product quality sub-par? **Can you afford not to take this class?**

We'll teach you how to get your products to market faster with fewer defects. Our recommendations are *practical, useful today, and tightly focused* on embedded system development. Don't expect to hear another clever but ultimately discarded software methodology. You'll also take home a 150-page handbook with algorithms, ideas and solutions to common embedded problems.

Here is what some
of our attendees
have said:

Thanks for a great seminar. We really enjoyed it! We're already putting the ideas you gave us to use.

J. Sargent, CSC

I like your practical, no nonsense advice backed up with numbers, your dynamic presentation style, and the nice handout that you gave us. I will definitely recommend your seminar to other programmers.

Ed Chehovin, US Navy

I just wanted to say thanks for a great seminar last week. Already the information you gave has proven useful – I used that ISR trick and we finally found an error we've been chasing for months.

Sandeep Miran

Thank you so much for a great class! Now my co-workers think I'm the guru!

Dana Woodring, Northrup Grumman

Did you know that...

- ... ***doubling the size of the code results in much more than twice the work***? In this seminar you'll learn ways unique to embedded systems to partition your firmware to keep schedules from skyrocketing out of control.
- ... ***you can reduce bugs by an order of magnitude before starting debugging***? Most firmware starts off with a 5-10% error rate – 500 or more bugs in a little 10k LOC program. Imagine the impact finding all those has on the schedule! Learn simple solutions that don't require revolutionizing the engineering department.
- ... ***you can create a predictable real-time design***? This class will show you how to measure the system's performance, manage reentrancy, and implement ISRs with the least amount of pain. You'll even study real timing data for common C constructs on various CPUs.
- ... ***a 20% reduction in processor loading slashes development time***? Learn to keep loading low while simplifying overall system design.
- ... ***few watchdog timers are properly implemented***? Most are partial solutions to a complex problem. We'll show you how to build an awesome WDT.
- ... ***most interrupt-driven timers are improperly coded***? Subtle asynchronous issues always lead to erratic timer reads and crashes. The solutions are not obvious, but easy to implement.
- ... ***reuse is usually a waste of time***? Most companies fail miserably at it. Though promoted as the solution to the software crisis, it's much tougher than advertised. You'll learn the ingredients of successful reuse.

Busy Schedule? . . .

If you can't take the time to travel, we can present this seminar at your facility.

We will train *all* of your developers and focus on the challenges unique to your products and team.

Thanks for the terrific seminar here at ALSTROM yesterday! It got rave reviews from a pretty tough crowd.

Cheryl Saks, ALSTROM

Thanks for a valuable, pragmatic, and informative lesson in embedded systems design. All the attendees thought it was well worth their time.

Craig DeFilippo, Pitney Bowes

I just wanted to thank you again for the great class last week. With no exceptions, all of the feedback from the participants was extremely positive. We look forward to incorporating many of the suggestions and observations into making our work here more efficient and higher quality.

Carol Batman, INDesign LLC

Contact us for info on how we can bring this seminar to your company

E-mail: info@ganssle.com
or call us at 410-504-6660

What are you doing to upgrade your skills? What are you doing to help your engineers succeed?

Do you consistently produce quality firmware on schedule? *If not . . . what are you doing about it?*

Better Firmware... *Faster!*

A one-day class
April 23, 2008 - Chicago
April 25, 2008 - Denver
May 19, 2008 - London

Spend a day with Jack Ganssle, Embedded System Programming's Technical Editor and columnist, and learn new ways to get your products to market faster, and improve your resume with the 0.7 Continuing Education Unite you'll be awarded.

Registration Information

All of this, plus 150 pages of handouts, for just \$695 per person. Plus you will receive a personalized certificate of completion at the end of the course.

Groups of 3 or more registering together pay only \$595 each.

Register early and save. Sign up a month in advance, and receive a \$50.00 discount.

Fax this form to 647-439-1454. Or, register by phone at 410-504-6660 or via email to register@ganssle.com.

Cancellations made more than 14 days prior to the class are refundable less a \$50 fee. Cancellations made within 14 days are non-refundable, but are 100% transferable to all courses we offer.

April 23, 2008

Chicago, IL

Hilton Oak Lawn
9333 Cicero Ave.
Oak Lawn, Chicago

April 25, 2008

Denver, CO

Embassy Suites Hotel
7525 East Hampden Ave.
Denver, CO

May 19, 2008

London, UK

Jurys Inn Heathrow
Eastern Perimeter Rd
Hatton Cross, Hounslow

Today's Date: _____ **Registration Form**

Name: _____

Company: _____

Mailing address: _____

City, State, Zip: _____

Phone: _____ Extension: _____

Fax: _____

Email: _____

Location: Chicago _____ Denver _____ London _____

Number of attendees: _____

Purchase Order Attached. P.O. Number: _____

Charge to: Visa MasterCard American Express

Card Number: _____ Expires: _____

Name on Card: _____

Signature: _____

Fax this to 647-439-1454. Or, call us at 410-504-6660.