APPENDIX D

HOW TO BE A SUCCESSFUL CONSULTANT

Introduction

It is not possible to predict just how the Environment design idea will spread through the industry. But I strongly suspect that, initially, at least, consultants will play a major role, both as designers of actual Environments and as teachers of the methodology. Since many of these consultants may have spent most of their working lives as full-time employees, I thought it appropriate to conclude this book with some guidelines on making the transition. (The guidelines do not merely apply to the transition to Environment design consultant, of course.) This transition sounds simple enough and, if one already has the technical skills, it appears to require nothing more than a little selling of oneself. As many who have become consultants in the documentation field have learned, this is not quite so. The main reason is that writers who emerge from years in a large corporation typically are unaware of the importance of local culture in the computer business, because they have been in the same culture for so long. By *local culture* I mean the knowledge, including knowledge of acronyms and buzzwords, and the behavior, that signals to others “This person is one of us.” A local culture can be centered on a language, e.g., LISP, or C, or an operating system, e.g., Unix, or a product type, e.g., data base systems, or a computer type, e.g., PCs or Macs, or it can be centered on some combination of these. My complete ignorance of the importance of local culture cost me two jobs when I first embarked on my career as a consultant.

The following is based on my, and other contractors’, experience in the San Francisco Bay Area, in the nineties. Writers in other parts of the country may disagree with some of the recommendations, of course. I have tried as much as possible to avoid repeating common knowledge, e.g., in how to conduct yourself in an interview.

Abilities You Need

To be a successful contractor, you need the following abilities:

- ability to estimate person-hours, hence cost, to do a job;
- ability to keep an honest record of the time actually spent on project(s). Normally, you don’t charge for commute time or car expenses unless this is a large proportion of time actually worked. You don’t charge for meal time. You don’t charge for time spent in non-business-related conversations with company employees. You

don’t charge for personal phone calls. You don’t charge for breaks if they amount to more than, say, fifteen minutes a day. Normally you will wind up charging about six hours per “eight hour” day;

- ability to use standard systems: DOS, Unix, vi, Emacs, MicroSoft Word, Word Perfect;
- ability to write clear technical English;
- ability to learn new products fast.

Learn the Legal Requirements of Contractors

In recent years, the IRS has cracked down on what it considers abuses of the designation *contractor* or *consultant*. Since the rules often change, and since they may vary from place to place, I will not attempt to summarize them here. But you should

find out what they are, either by asking for a source at the reference desk of your local library, or by looking them up in the professional journals, e.g., the publications of the Society for Technical Communications (STC), or by asking an experienced contractor.

Not all of this government intrusion is purely to make your life difficult. Part of it is aimed at protecting you, since some companies have attempted to use the contractor designation as a way of hiring employees without providing them any health benefits and without paying various payroll taxes. If you are an *employee*, then every aspect of your work is, or can be, controlled by the company. If you are a *contractor*, on the other hand, then you determine how the job is to be done, and you deliver the promised result at a specified time. An employee works at one of the company’s office sites. A contractor often works at home or at an office outside the client company. (A person calling him- or herself a contractor may have to prove this to the IRS.) It is illegal for a company to treat a contractor as an employee.

A contractor has certain obligations, of course. One is that he or she must pay income taxes regularly, e.g., quarterly, whereas an employee has this done by the company.

Selling Yourself

The need to sell yourself, day in, day out, is probably the most important single test of whether a person is cut out to be a contractor or consultant. After a lifetime of hard work in one's profession, it is not easy to be told by a twenty-year-old gum-snapping receptionist that you should "send a resume to our Human Resources department." (Avoid Human Resources and Personnel and Staffing departments at all costs, unless you are already an employee of the company and are simply interested in changing jobs, in which case these departments can be useful.)

Always try to find out the name of Technical Publications Dept. managers and of members of these departments. Develop a list of names and companies, keep copies of letters you send. If someone says that they are not hiring right now, ask when they think they might be, and be sure to call them back at the specified time.

Make calls every day, even when you are working. When you are not working, no matter how bad you feel, set aside at least, say, an hour each day to make calls. You will be surprised to find that someone who didn't even respond to a letter and resume you sent, will call six or nine months later.

Offer a commission — say, 10% of your first month's gross income — to any fellow professional who gives you a name that leads to a contract. (This has worked well for me.)

Perhaps most important, try to find a local superstar in your field who knows the employment territory, and offer to pay him or her — *by the minute* if necessary — for advice and leads, plus, of course a commission on any work the person leads you to. This has been the best job investment I ever made.

Interviews

If you get an opportunity to be interviewed, take it. Always go to interviews. You have nothing to lose and something to gain.

Countless books have been written on how to present yourself in an interview. I will not repeat that material here, except to say that you should always dress to impress your interviewers that you are a serious professional. Do this even if you know for a fact that every one comes to work in jeans or shirts and T-shirts. For men this means sports jacket, tie, shirt, pants, shined shoes. Sit erect, avoid signs of nervousness, e.g., repeated movements of hands and fingers. *Always* be on time.

Show yourself to be positive, eager to apply your considerable knowledge and skills to helping the company make money. Show interest in their product. Here, asking questions is good. “Tell me about your product!” If you have good reason to believe your prospective employer is genuinely interested in having an Environment designed, then do a few top-level screens, on paper, from available company literature. “When a user presses ‘Help’, he or she would first see a screen like this... Now suppose he or she decides they want to do ... Then, clearly they would click ... which would immediately bring up ...”

Interview the interviewers about the job — in particular, *interview your prospective manager to be sure that he or she is someone you will be able to work for.* The single most important guideline that I have found (through hard experience) is the language-awareness of the person. Do they seem unconscious of their speech? Do they make grammatical errors? Do they demonstrate little or no concern about choosing the right word when that is important? Do they seem to believe that knowledge of buzzwords, acronyms, and product names is a mark of intelligence or a mark of understanding the realities these terms represent? Most important, do they take minutes to say what could be said in seconds? Then chances are that their thinking will be equally muddled and you are better off looking elsewhere.

Once you believe that your prospective manager is someone you will be able to work with, then impress them with your thoughtfulness, care, unwillingness to accept just anything. No matter how desperate you are, make it seem that you are in demand, that their job is just one of several you have interviewed for.

Warning: Beware of prospective clients who are out to get what amounts to a free design for their documentation, so that they can then go out and hire much cheaper labor to implement it. Initially, your only task is to get the job, to convince the prospective client that you have the skills and knowledge that will benefit them.

Never present an impression of Do it my way or else. This is particularly important for readers of this book who no doubt will be dismayed by the backwardness of the thinking of prospective clients regarding the whole question of documentation. Strive to give the impression that you are a person who knows many ways of approaching a problem but that, after presenting the alternatives, you are always glad to abide by the company’s decision. Make your suggestions, then acquiesce. It is perfectly acceptable to say things like, “Well, some companies are starting to do” such-and-such, if it is in fact true. Make it clear that you are there to serve the company.

One way to cut down on the chances that they are merely using knowledge of the term *Environment design* and a few related concepts as a means of locating better-than-average temporary workers to do old-fashioned documentation is to charge a much higher rate than such workers would normally get. Be frank about this if you are questioned. Tell them that you are offering them a documentation product that will increase their sales in the marketplace. Tell them you are perfectly willing to guarantee your work (as described in chapter 4), which is more than the vast majority of writers can do.

One warning I must make is that if you find yourself just looking for a job, whether or not it has anything to do with Environment design, then beware of appearing to be too intelligent. You should not appear to be a person with a mission, e.g., that of promoting new ideas. Except in the rarest circumstances, you should never give the appearance of being a thinker.

The classic example of the consequences of ignoring this advice happened to a friend who has been working with me on the ideas in this book. In this case, he was applying for a full-time position. Upon being told by the interviewer what documentation problems the company was having, this friend launched into an inspired pitch for doing things the new way. He also revealed that he had several years of programming experience and an advanced degree in computer science. When he was finished, there was a long pause, and then the interviewer looked him in the eye and said, "Well: you speak in grammatical sentences, you have a fast mind, and you're obviously creative: I don't think there's a place for you here."

Demonstrate knowledge of the culture (see below).

Never say, in response to a question about whether you know a specific product or technical area, "No, but I can learn," still less, "No, but I'm sure that I can learn it," and even less, in the case of a product, "No, but that's the kind of thing one can learn in a few days, and I'll be glad to work at no charge until I have demonstrated I have learned it." These replies, for reasons I have never been able to figure out, immediately brand you as a loser. If you know enough to get by, say "Yes." If you don't, say "No" and let the interviewer volunteer the suggestion that perhaps it might be possible for you to learn it.

Avoid fixed price bids on developmental projects. It is still notoriously difficult to predict how long a software project will take — or, more precisely, predictions are almost always too optimistic.

Before You Accept the Job

Before you accept any job, *get a contract*. I don't mean a legal contract, I mean a contract for what you are to deliver by when and in what form. If a prospective client says, e.g., that they want you to spend two or three months "learning the product," explain, in the politest possible terms, that that is not how you go about producing documentation, that you cannot risk accepting such a job unless the client is willing to spell out, in detail, what he or she will consider "having learned the product."

Be extremely wary of projects which are described as "just something informal for internal use." Since it is almost certain that you will not be able to read the minds of your clients, you are almost certain to deliver something far removed from what they expected. These projects more than any others demand that you ask the client to spell out in detail the typefaces, formats, language style, and structure of the document. If the client looks down on you for not being able to read his or her mind, then politely decline the work.

It is important to specify not only the form of what you deliver — typeface, format, outline, on which size of floppy disk — but also the form of review versions. Do not assume, e.g., that you can embed questions in the text itself. Some clients become uneasy when they see anything that doesn't look "finished," even when you have explained that you have simply indicated questions which you will have to be answered before the document is complete.

The contract should also specify how much time you will be allowed *each day* to talk to other engineers and programmers. Half an hour? An hour? Emphasize that you will assume that the engineers and programmers will have been informed of this obligation when you begin work, and that you must reserve the right to ask any questions which seem important to you, regardless of how naive or unnecessary they may seem to the engineers or programmers. Explain that the per day figure is cumulative: if the figure is one-half hour, and you do not bother anyone for, say, four days, then you still have the right to bother them for two hours — not necessarily all at once — later on. Be absolutely sure that the client understands this. I know of several contractors whose reputations suffered because they did not make this clear at the start of contracts, the clients feeling that the contractors "wasted" too much of their employees' time with questions. If the prospective client seems unwilling or uncertain about making the time commitment, decline the work. Make it a rule to consolidate your questions so that you only have to bother engineers and programmers once a day, rather than two or three times. Always make it clear to them how efficient you are trying to be, how much you are trying not to consume more of their time than is absolutely necessary.

Project the Right Image

The image you should always try to project is that of a quiet, hard-working, totally honest professional who knows the tools he has been hired to use, where *tools* here includes operating systems, and who learns the company product very rapidly. Contract work should not look like a learning experience, even though it often is.

You should *always* meet deadlines if at all humanly possible. Even if you have to work a few hours without charging. "He meets deadlines" is one of the most important recommendations you can hope to have.

Learn the Culture!

No matter how intelligent you are, how ambitious, how committed to doing good work, if you don't demonstrate knowledge of the culture in which you work, your days will be numbered. Knowledge of the culture is demonstrated by knowledge of the meaning of, and correct use of, buzz words and other technical terms, by liking and disliking the right things, e.g., disliking all the other programming languages in a given programming language culture, and knowledge of the use of equipment or software, e.g., Mac, PC, Unix. Also knowledge of the names of the leading companies, executives, gurus, and other personalities in the culture, and of recent history: which companies were on top in the recent past, their recent product history. I know of technical writers who have gotten contracts purely because they spent an hour or so before the interview learning ten or twenty of the most common technical terms used in the work. Beyond that they were completely ignorant of the subject.

The main question on the minds of those doing the hiring, as well as those with whom you work, is "Is this person one of us?"

Personal Work Habits On-Site

If you work in the company's office, keep personal phone calls to an absolute minimum. If you make more than, say, five minutes of personal phone calls in a day, do not charge for this time. Charge all personal phone calls outside your area code to your home number. No exceptions. Speak in the quietest voice you can, so that no one can accuse you of disturbing their work. Avoid annoying personal habits, like talking to yourself about what you are doing — "OK, OK, let's see now, we've got that, and got that, so now if we could..." — and sighing and groaning . Don't crackle the paper of your junk food, don't snap your gum, don't chew food loudly. It is absolutely amazing how annoying these minor disturbances can be to other workers, particularly when their own work is not going well.

Don't ask questions of the person you report to or anyone else who has the power to influence your continued employment, re-hiring, or recommendations, unless you are absolutely certain they will not look down on you for doing so.

It is often necessary to learn a lot fast about a software system. You should cultivate friends or professionals whom you can call on, if necessary for a fee, and find out what you need to know.