PROFESSIONAL COLDFUSION CONTENT FOR THE COLDFUSION PROFESSIONAL

The Fusion Authority Quarterly Update

Vol. II Issue I

SPECIAL BUSINESS ISSUE:

Business practices and concerns to channel your programming skills into cash

How to Avoid Unpaid Consulting

Making Google Pay

Deconstructing the Consulting Contract

Plus — Developing Applications with Tranfer

And Much, Much, More...

VOL.II ISSUE I $14.95

ISSN 1932-0264

21>

9 771932 026000

CONTENTS

Fusion Authority Quarterly ■ Vol. II ■ Issue I

Deconstructing the Consulting Contract

By Jeffry Houser

I sat at a table in a private booth in a dark corner at the back of a restaurant. Sitting around me was a group of programmers, application architects, and project managers. We were all listening to a single man, Mr. Venture Capitalist. We didn't have an idea to sell him, but we just knew that with the proper idea, we could finally move away from the drudgery of work-sleep-eat to the joys of work-work-work.

"Let's start brainstorming," said Mr. Venture Capitalist, as he took a bite of his meal. We went around the table one by one sharing ideas as the VC munched. As the conversation bounced around the table, I was finally able to get a word in. I don't remember my exact words, but Mr. Venture Capitalist cocked his head slightly and gave me an odd look. "Oh, don't worry about Jeffry," our leader chimed in. "He's a closet lawyer."

Why a Contract?

I was outed as a closet lawyer. Before that moment, I never thought of myself that way, but as a small business owner I've made every mistake you could possibly imagine, and then some. I've learned quite a bit along the way about business, client relations, and most importantly consulting contracts. This article is going to impart some of my knowledge and experience to you. Before you read further, you may want to peruse the consulting contracts I use. I've went and posted them to my blog at http://www.jeffryhouser.com, and they can be downloaded from the business pod. But, be forewarned, I am presenting these as examples for education purposes only. You'll have to talk to a lawyer to find something that will work specifically for your situation.

Why would you want to use a contract? There are plenty of reasons. First, you want to be seen as a professional. Professionals use contracts! Amateurs rely on handshake deals. When you present a contract to a client, you are telling that client that you are a professional who takes his business seriously. Second, putting your agreement in writing will make sure that both you and your client think the process through. We all know from our software development experience that the more time you spend planning a project, the less time you'll spend on execution and the more effective that execution will be. Well, the same goes for contracts. The more you have written up front, the less likely your client is to be disappointed with the results, and the less likely it is that you'll have disagreements down the line. Third, and perhaps most important, you want to cover yourself just in case things do go horribly wrong. In your life as a consultant, at least one project will go horribly wrong.

These are some of the important items you want in a contract:

- Project Scope: What are you going to do during the term of this project?
- How do you get paid: Are you getting paid hourly? Or do you have a fixed fee for the full scope of the project? When do you get paid? Will you receive an advance? Will there be a termination fee? What happens if you don't get paid?
- Timeline: When does the project start? When does it end? Are there milestones that must be reached?
- Code Ownership: After the project is completed and the bills are paid, who will own the resulting code? Can you reuse the code in future projects? Can they reuse it in future projects? Can you demonstrate the code / project to potential clients or use it as proof of skill?
- Termination clause: What happens when things don't work out? The contract should let you get out of a bad client relationship.

I was once told that a contract never dictates a person's behavior. It just gives you the right to sue them later. When you wind up in court over a breach of some sort, you'll be very happy that you have a well-written contract. The daytime court TV shows are littered with people whose problems stem from not signing a contract. It makes for good entertainment, but not good business. With the above items worked out up front, both you and your client can enter into the agreement knowing what is expected and that all bases are covered. The remainder of this article will look at these items (or clauses, in lawyerspeak) in further detail and discuss potential negotiating strategies.

The Non-Disclosure: Don't Share our Information!

The first time you meet with the client, there is a chance you'll be handed a non-disclosure or non-compete agreement. Collectively, this document or series of documents is referred to as a Non-Disclosure Agreement, or NDA. An NDA isn't a guarantee to hire you; they just want to cover all their bases and prevent you from looking at their code and becoming a direct competitor next week. Every client thinks that their ideas, procedures, and approach are unique to them and a valuable business advantage. Most of the time they are wrong, but I don't knock them for wanting to protect what is uniquely theirs, such as their client list, for example.

Here are some items that you'll probably find in a non-disclosure agreement:

- Definition of Confidential Information: What is considered confidential? The agreement should specify the type of information you cannot share. The client is going to want to make this as broad as possible. "Confidential information shall include, but is not limited to [insert huge list of everything under the sun]". In most cases, clients really want to prevent reuse of their code and client list, but you should read that list carefully and use your own discretion. "Not limited to" is vague and means anything at all could be included, whether specifically listed or not. If you post their QuickBooks records onto a newsgroup, most likely they're going to be upset, even if "financial records" isn't specifically listed.
- What isn't Confidential Information: I've seen some agreements that specify what isn't considered confidential information. Usually this includes information that is public domain or information that the company chooses to make public. If you can read about it in a press release on the company's web site, there is a pretty good chance they won't come after you for releasing the information.
- When can you share confidential information: When the FBI comes knocking at your door asking for information about your client, you don't want to be in violation of an NDA by sharing information. Usually you can share NDA information with employees or subcontractors of your company, as long as they are bound by the terms of the NDA. If you have employees who will be doing the work, or subcontractors who will be helping you with the job, make sure that you can share with them the information they need to do the job.
- Return of Confidential Information: Most agreements have a clause that allows the party to ask for the return of confidential information. Sometimes this clause says destruction is acceptable in lieu of a return, and destruction is a more common approach for digital files. I've never had a situation where people have used this clause, but have no problem with it being in there. Usually you have seven days to return or destroy. When destroying (read: deleting) digital files, you must be careful to address any backups you potentially made.
- What happens when the agreement is breached? If you accidentally leak confidential information, you're screwed. The contract will say this in lawyerspeak, but generally you are liable for significant sums in damages.
- Length: The span of time in a non-disclosure agreement is usually vague. The contract may have an expiration date, but that date is usually meaningless. You don't want to be the one to release confidential information. It's safest to keep your mouth shut until the information is freed by someone else.

As you sit with the client and read the NDA agreement, you should ask yourself, "Is this a mutual NDA?" Is the client going to protect your confidential information in the same way that he protects his own information? Or can he start soliciting your clients the day after you provide him with demos of past work? I always push for a mutual agreement. If that isn't possible, I make a client sign my own NDA before I provide him with any confidential information from DotComIt.

The Non-Compete: Don't Work with our Clients!

A non-disclosure agreement is often combined with a non-compete agreement. I have seen cases where the client gives me one, or the other, or two separate agreements. Most of the time they are combined. The non-compete agreement can take a few different forms. The form I prefer will prevent you from working with your client's customers. They don't want you coming in, doing a project, and then walking away and taking their current customers with you. That is a perfectly acceptable concern. Sometimes the non-compete will prevent you from using their confidential information (see previous section) to help service your own new or existing clients. I don't have a problem with that either.

Sometimes the non-compete will prevent you from working within the same industry. I understand why companies would ask for this. They don't want to train you on their dime, and then have you go work for a competitor, or become a competitor. I always feel some caution about this, though. I once refused to sign an agreement that would have prevented me from working "in their industry or similar industries." It was a web development company that wanted to subcontract work to me. I thought this was too vague, since all my current, previous, and future clients would violate this. They were unwilling to negotiate. The situation seemed doomed and I walked away.

Non-competes often have a clause that will prevent you from approaching employees for business outside of the company. As with client clauses, I don't usually have a problem with non-compete clauses. I have seen one that was retroactive, meaning that by signing I was guaranteeing I hadn't solicited employees or clients for some given period of time before signing. It was a catch-22 because without seeing the client list or employee roster, I couldn't actually guarantee that I had never worked with or solicited their clients or employees. I was able to negotiate the retroactive scope of the clause out of the contract before signing.

As with the non-disclosure, make sure that a non-compete has a time frame. Usually they are in effect for one to three years after you complete the project. People change jobs and companies change vendors all the time. You never know when you're going to run into a former employee or customer of your client. I think one year is reasonable. Three years is a little excessive.

One important point I'll note is that I always ask for the non-disclosure or non-compete agreements before my first meeting with the client so that I can review them and, if needed, send them along to my lawyer. If the client wants you to sign such agreements, but is uncomfortable with you having a lawyer review them, then that should be a red flag.

The Master Contract

After all the initial paperwork, you talk to the client and they show you the project and you decide to move forward. People generally hire me in one of two different ways: either on an hourly basis or on a fixed-fee project basis. I'll explore some of the differences later in this article, but there are common elements in both types of contracts. I'm going to discuss those elements here.

- Introduction: The introductory part of the contract usually defines who you are and who your client is. It will specify that each is a sole proprietorship, LLC corporation, or some alternate corporate entity, and the intent of each entity. This is usually a short statement along the lines of "DotComIt is in the business of providing technical consulting and client is interested in hiring DotComIt for said services." Often this part is dated, and this date represents the contract creation date, which may be different than the contract signing date.
- Definitions: If any definitions are needed, they are listed here. My default consulting contracts don't list any definitions, but if you take a look at any software license you'll probably see a lot of definitions. These are the various words or acronyms used in the contract and their meanings.
- Relationship of Parties: This type of clause usually states that each party is a separate entity and that the contract does not create an employee-employer relationship in any way. I like these clauses, and don't usually have any problem with them.
- Scope of Work: This defines what will be done as part of the project. In a fixed-fee project bid, the scope usually contains an in-depth description of what will be created, sometimes with screenshots

of prototypes. In an hourly contract, this is usually a generic line that states "Web Development" or "ColdFusion and SQL Server" or some such thing. I have set my contracts up so that there can be multiple projects, AKA work assignments, under the same master consulting agreement. This seriously cuts down on paperwork for repeat clients and easily allows for situations where the client hires me on a fixed fee for initial development, and hourly for minor maintenance.

- Payment: How will you get paid? Via check, credit, or PayPal? When will you get paid? Weekly or monthly? Will there be late fees? Put all this in the contract. If you don't use your own contract, you may have to ask for such a clause. Usually it can be added without a problem. Some people question late fees. To counter that, you could offer a lower late fee, or a longer payment term – forty-five days instead of thirty days, for example. If all else fails, offer to remove the late fees altogether. I'm usually willing to do that if the client is willing to provide an advance. If you do a lot of subcontracting work, you may see a "You get paid fifteen days after we get paid" clause. I hate these. I can't budget for "fifteen days after you get paid." I often ask what the payment terms are between my client and their customer. Then I ask for a payment term of fifteen days after that. So, if my client gets paid in thirty days, I ask for a forty-five-day payment term. If that doesn't work, I'll try to add a clause that says "no delivery of project until payment is received." Sometimes I may let this slide if the client is willing to provide an advance. Most of the time, if the client is unwilling to change his "You get paid fifteen days after we get paid" clause, I end up walking from the contract.
- Collection: If you have to get a lawyer involved to get paid, your contract should allow you to recoup the costs of the lawsuit, which can eat up a huge chunk of the monies owed you.

  If a clause allowing you to recoup collection costs is in a contract that you bring to the negotiating table, they probably won't question it. If you ask the client to insert such a clause into a contract that they've handed to you, they will probably fight it, though I'm not sure why. If they fight it, you can ask, "Do you plan to pay me on time?" The client will of course say, "Yes." Whereupon you can respond, "Then this clause will never come into play, but it sure would make me more comfortable moving forward," and the client will probably concede.
- Termination: A good contract will give both you and your client the right to terminate, usually for any reason. If the client wants you to sign a contract that does not give you the right to terminate, but does give them the right to terminate, ask for a mutual termination policy. You'll probably get it. If your client terminates, make sure that you will still get paid for the time you have already worked. Sometimes you can get a termination fee built into the contract. I turn down jobs all the time based on commitments I have made to current clients. The termination fee is designed to keep you cashflow-positive until that next opportunity comes around. In hourly projects, where the client makes no commitment to provide work, I'll offer to waive a termination fee. They aren't promising me work so I'm not blocking out time for them. If the client is committing to my time, I chunk it out for them and turn down projects based on those commitments. Often I make any advances or monies paid non-refundable.
- Term: Each project should have a start and end date. No one complains about this.
- You must re-write broken code at no extra cost: I've seen contracts with the vague clause that any work not built to "spec" must be re-written at no extra charge. If this is on a project or fixed-fee bid I don't mind it in there. The client is committing a certain budget to a certain feature specification and I am committing to build code that meets that specification. If there are bugs, then I clearly messed up. I'll fix them at no extra cost. In hourly contracts, I fight this tooth and nail. Generally on hourly contracts the specs can change multiple times a day. I'm getting paid hourly because time was not taken to figure it out from the beginning. The client wants X, then changes their mind to ask for Y, and then goes to ask for Z. And then the big boss man looks at it and says, "Why didn't you implement X?" This is not a situation where you should be making all the changes for free.
- Code Ownership: Most likely the client will want to own the rights to everything you do as part of the project. I always try to add a clause that allows me to demonstrate the project for marketing purposes or to show proof of skill. This works for almost all clients except those who place high importance on the proprietary elements of their application. (I once worked for a client who had a web form that submitted stuff to a database. Don't tell anyone, though, because I'll get sued).

- We own your thoughts: I once turned down a job because the contract stated that the client would own all my ideas, creations, patents, trademarks, or any other intellectual property whether they were created on company time or not. As a songwriter as well as a programmer, business owner, and closet lawyer I felt this was vague enough that they could legally claim ownership of my musical creations, even though the job was a programming job. I would never agree to such an overextending clause. This could also apply to the next product idea you want to build. Thankfully, few people ask for that when hiring consultants or outside vendors. If you're an employee, be aware of your employer's policies on such things.

These are most of the major issues that you'll deal with when entering into legal agreements for software development. For the rest of the article, I'm going to discuss some of the elements that are specific to working either hourly or on project-based flat fees.

Working Hourly vs. Fixed Fee Projects

Some clauses come up in hourly contracts, but not in fixed fee project bids. Let's explore some of those elements.

The first thing that comes to mind is some form of approval process. When I first started consulting, I did not have an approval process in place, but I eventually added it. That service the client desperately needs now seems less valuable once you invoice it. The further you get away from the event, the foggier the client's recollection of the work you performed. If you're onsite on January 1st trying to fix a server problem, and then invoice for that time on February 1st, the client will probably contest the ten hours you spent.

- "I don't remember you here for that long."
- "You were gone for lunch for over two hours that day."
- "You should have fixed the problem in 4 hours, not 10."
- "We didn't have that much money in the budget for this problem."

I've heard all the excuses. My eventual solution was to introduce a weekly time sheet procedure. Many temp agencies use this process, with a representative from the client approving each day's work. Instead of doing it daily, I do it weekly. Once a week I send the client a time sheet displaying the hours I put in, the day they were put in, and the tasks that I completed. When invoicing time comes, and the client complains, I'm able to wave the time sheets in front of them saying, "But you already approved my time." This has worked well in the cash flow department. I generally will not start work on "week 2" until "week 1" has been approved. This way, if I have to eat it and suck up a loss, at least I'm not eating a whole month.

On fixed-fee projects, I'll have a "phase completion sheet" instead of a weekly time sheet. The client has no idea how much time I put in. They generally care about results and deadlines, not hours. The phase completion often comes with some form of additional payment, depending on the length of the project. Both time sheets and phase completion sheets have a dual purpose. In addition to the "cover yourself" aspect of billing, they also help to keep communication lines open with the customer. When the customer sees these, they know that the project is moving forward, and exactly what is being done. If there are problems on either end, they can be addressed in a timely fashion.

Often I'll include a "deliverable" definition in the contract. After the project is done, what are you providing to the clients? In most cases, this is simply "source code files and related development documentation." It helps clients to know what their end product will be before going in.

Requirements change all the time. In hourly contracts, this is great because the clock just keeps on ticking while you continue to work. In fixed-fee bids, this is not so great because you can be sucking up a loss while you change the application. To help make the client aware of changes, I've added a change process into the mix. If there are changes, I write up a change form and pass it on. The change form encompasses the change of spec, the change in timeline, and change in cost. There is always some padding built into the project cost, so the first change will often not cost more. But, at least the client is aware that they are changing the agreement. It makes them less likely to fight it when future changes do create extra cost.

Conclusions

If at all possible, I always ask my client to sign my own contracts, but often clients have their own consulting contracts. It's a small miracle that I was able to keep my business afloat during the early years, since I was unprepared for the possibilities that abound in the small type. I learned about due business process the hard way, but now you won't have to. I hope that reading through this prepares you for what you'll find the next time you're handed a contract.

If you didn't already look at them, don't forget that I posted the contracts I use on my blog at http://www.jeffryhouser.com. But, remember, they are for educational purposes only; you really need to talk to a lawyer to find something that works for you.

Jeffry is a technical entrepreneur. He has a Computer Science degree from the days before Business met the Internet. He worked for a business-to-business consulting firm before eventually starting DotComIt, a web consulting company. He is co-manager of the Hartford Adobe User Group, author of 3 ColdFusion books, a frequent contributor to that other CF magazine, and has spoken at user groups and conferences all over the country. In his spare time he is a musician, old school adventure game aficionado, and recording engineer. You can read his blog at www.jeffryhouser.com