

The Journey: Phase II - The Clone Wars

Posted At : September 5, 2012 12:42 PM | Posted By : Mark Kruger

Related Categories: Business Of Development

I was recently chastised by a twitter follower (a beat down in 140 characters or less) for starting series that I fail to finish. So I'm coming back to my "Journey" series to add to the CF Webtools story a bit. When last we met on the subject I spoke of the **3 attitudes** you need to succeed in the consulting business:

- Work Hard and Be Patient
- Be a People Person First
- Avoid Perfection Paralysis and Do What You *Can* Do

With those thoughts in mind I'd like to talk about Phase One of your consulting business - building Clones.

More of Me

Anyone who's ever been successful as a contractor and thought about expanding has thought to themselves, "If I only had 2 of me." Aside from the obvious stress it would put on my wife you would think that having 2 of the Muse would be exceedingly useful. But knowing me, I would doubtless be playing golf right now leaving me behind to do all this work. That's just like me. It would make me so angry I'd be beside myself. Still, the idea is compelling when you are starting out - so compelling that you think about it a great deal when contemplating that all important *first hire*.

Consulting businesses are often started by knowledge experts with little or no business experience. When expanding such a business the first choice is usually "more of the same". In my case since I worked a certain way, I geared all my documentation, proposals, and estimates to the skill set of the Muse. So what did I look for in my first hire? Muse II of course (same level of action with a weaker plot I guess). It made sense to expand the current way of doing business by simply gathering similar skill sets to myself and dividing the work up amongst them. My first hire (Jason Herbolsheimer who is now CF Webtools VP of development) was an energetic can-do programmer able to find creative solutions to difficult problems. He worked at a similar speed to my own and was (and still is) a terrific people person. It was a great fit. Suddenly we were able to do roughly twice the work as before. In fact, my first 3 hires were like that. They were proven CF developers who I had known previously. Two of them had worked with me at my previous Job. The 4 of us divided up our customers and simply worked them in the same manner that I had worked them when I was an individual contractor.

This approach reminds me of that moving company "2 men and a truck" (would that be a "Mac" truck?). My guess is they started out as 2 men... and a truck. When they decided to expand they were probably considered changing their name to 4 men and 2 trucks, then 6 men and 3 trucks. There's some magic to this approach. It actually works well in many cases - especially if you assemble the right folks. If your team members work well independently and have the right soft-skills (inner-directed, owning problems, eclectic skill set, customer driven etc.), it can work *quite* well. The 4 of us did fine and had a great time along the way. I know of 3 or 4 consulting companies who operate at this level and intentionally stay at this level. And why not? They make good money, have very low overhead, and the level of responsibility is less crushing. Still, if you plan to expand beyond a handful of developers, the "clone model" (not to

be confused with cloning an actual model which my wife says is out of the question) comes with some penalties.

Founder Syndrome Death of Innovation

Having 4 clones doing similar work means lots of self-congratulatory back slapping. We were a homogenous group of Nebraska men and we were all doing roughly the same job. Other than Ryan (who has definite opinions about where to eat) we even liked the same restaurants. This sort of environment was great for my self-esteem and made going to work a pleasure. We all liked ColdFusion. We all had similar ideas about technology. We all approached customers in a similar fashion. Conflicts were pretty rare and we managed to survive and thrive.

Or so we thought... The truth is that a staff of clones comes with built in blinders. Growth (personal and professional) slows when you lock yourself in to tried and true points of view and fail to keep striving to see a bigger picture. Conflict - professional conflict, not the kind you are used to at the Holidays - is healthy and necessary for growth. In genetics, when a population of species is founded by too small a number of animals the lack of genetic variation leads to specific kinds of drift and eventually to deficiencies in subsequent generations. Now I'm not suggesting that consultants breed when placed in close proximity on a homogenous staff (and notice how I'm carefully avoiding the term "inbreeding"). But I *am* suggesting that a lack of variety within a staff keeps you from seeing things clearly and it will keep you from examining alternative solutions to problems you face. In short, it leads to a lack of innovation - a death knell for a development shop. And this is what began to happen to us. We bogged down in several areas.

Part of why this occurs has to do with control and comfort level. As a type A alpha male I am wired to attempt to control things around me. This makes starting a company a natural outcropping of my personality - after all I get to control every decision right? But as Uncle Ben said, "With great power comes great responsibility". I had to recognize that I was not equipped to make every decision. I need to see that my *personal brilliance* did not mean I was good at money, good at time management, good at hiring etc. So I had to make a *conscious effort* (still ongoing) to relinquish control of things within my company that I was less equipped to do than others I was adding to the team. This meant leaving off writing code, allowing input on hiring, focusing on business development and letting others handle day to day operations etc. This was (and still is) hard and requires sacrifices on a daily basis. But it is a necessary step for growth. That's not to say I'm not involved in the decision making process - but it *is* to say that I'm required to demonstrate trust, act in humility, receive knowledge when offered by those more learned than I etc.

One more problem comes along with the founder's syndrome. It has to do with the projection of your company image. Most founders start out projecting *their own* image - their knowledge and expertise, their accomplishments, their energy - and use the company as a sort of framework to magnify this projection. This was certainly true of CF Webtools and still is to some small extent. But to grow the company has to outgrow your own reputation. To put it another way, potential customers have to shift from analyzing your reputation and services to analyzing your *companies* reputation and services. This shift can't happen as long as you are still projecting your personality through the lens of the company. At some point the company has to take on it's own life and personality - of which you are a part.

Undervaluing Management Roles

As we grew beyond the 2 men and a truck phase (or maybe 4 guys and a mac), we noticed a certain raggedness to our customer effort. As new developers were added (with new skill sets and points of view) things did not go so smoothly. Sometimes developer's weren't prepared for the tasks assigned and our "sink or swim" model left them hanging out there in the wind. We also expected a very broad approach from each developer because that's how the Muse rolls. We expected them to interview the customer, develop requirements, shake out new features, participate in QA, manage their deadlines etc. But this level of ownership - suitable to an individual contractor - doesn't fit every developer. We needed to be able to leverage the exceptional talent we were able to hire without beating them to death with all the ancillary management details that come along with projects. Then too, projects began to bog down because decision making was now divided among the team - rather than being "owned" by a single individual. We were good at individual development but not so good at functioning like a team.

Cowboy Development

During the salad days (even though in an ironic twist our office is next door to Burger King), we took projects of a certain size. Most of our customers had small to medium size web sites providing a service or selling something and our role was to maintain them - adding features, doing upgrades etc. To this end we had a development server with the code hosted on it. It was rare that more than one of us was assigned to a particular customer. So it was easy for us to simply code on the dev server and then deploy to prod. Each of us had our own idea of what a good dev environment was like. I was a Homesite+ guy. I still use it as a light weight file editor and for browsing files on the network (although Sublime is creeping up on me - very chic). This sort of "1 to 1" relationship we had with our customers was fine up to a point. But as we grew we needed to expand not just our group of developers but also the size of the projects we were doing in order to keep pace with our business plan. So we aimed higher and picked up some larger projects.

Unfortunately some of these projects did not go well and we soon found ourselves backtracking and defending ourselves and scratching our heads about what went wrong. We needed a more "team oriented" approach but we were not equipped (yet) to work together in teams. We also saw pretty quickly that our rates were too low to support such projects. There's an exponential increase in expense with team development. Peer review, requirements, division of labor etc. - all began to take a noticeable slice out of our estimates. Meanwhile, keeping track of hours so we could bill was becoming more difficult. We had modest systems in place and our developers were instructed to log hours as they developed so we could create invoices matching our agreements. This was not working out so well. Examining our logging showed us that our developers seem to be under-billing somewhat - or often *forgetting* to bill. This should not have surprised us. The further away from the sales and financial end of a business the less crucial these tasks seem to be. Developers entrenched in coding have a hard time making things like billing a priority - and our system did not enforce a billing routine for them.

Role Confusion

As our company reached puberty there was the accompanying angst among the team and myself regarding our roles and function. Remember that we had started as a sort

of "band of brothers" - none of us really "over" the other and all of us pulling together. The Muse functioned as a sort of technology mentor at times due to his skill set but beyond that I didn't really feel like a "boss" or "manager". We all knew what we needed to do and pulled together to do it. We trusted each other without any real controls on our company life. But the downside of this is that there are no real clear lines of demarcation - no roles that we are bound to honor within the company.

To a degree this meant that decisions flowed through my office with no filter. I was responsible for vacation times, rates, billing frequency, payroll issues, salary, job assignments, bathroom paper supplies, project requirements, estimates - and of course I was continuing to code for customers full time. Jason (my vice president) was equally burdened as our staff crossed over 12 or 14 developers. At one point we had 14 people developing and none of us with a pure management role. Moreover, there was no clear communication within the company about who did what. As a short sited developer (and still cloning) I saw employees as "revenue producing" and "non-revenue producing" - and I wanted to be lean and mean, and not clutter up the staff with *management obstacles*.

No Sales No Service

Finally, we undervalued sales. It will come as no surprise to Muse readers that I'd rather be set upon by a pack of Mastiffs than have to spend time speaking with someone trying to sell me something. The last time I purchased a car I intentionally chose the one from the salesman who followed my admonishment to "only answer questions I'm asking" - a polite way of saying "speak when spoken to". This and my utter lack of modesty are my only flaws. So it should come as no surprise that I waited a long time to hire a salesman. Indeed our chief source of sales was simple word of mouth for the first 8 or 9 years. I did not know *exactly* what a sales person could bring to the table. In my mind I envisioned streams of gaggable buzzworthy marketing content combined with cold-calls and email campaigns.

Meanwhile under the heading of "That which I feared most had come upon me" I was doing sales work every week. I was writing proposals and estimates, talking to prospects on the phone, negotiating contracts and rates and generally getting my hat handed to me in most cases (while blissfully saying "thank you!"). Still, I could not bill for the time of a salesperson so this theoretical person was a "non-revenue producing" employee in my view. This was extremely short sited of me as you will see in my next post.

Winning the Clone Wars

So we sat at the cloning phase trying to make it work and we began to realize that our structure was out of synch with our size. In my next post (which is already written) I will discuss how we resolved or are attempting to resolve these issues as we struggled to free ourselves of our clone cocoon so we could emerge a beautiful butterfly of company - or at least move on to the next phase and the next set of challenges to our company life and culture. So in my next post we will talk about *recovery* from founder's syndrome (I'm a founder, but I can change, if I have to, I guess), valuing sales, adding diversity, building management systems that make sense, and reigning in cowboy development. Stay tuned.