

# Startup Without Falling Down

by Philip Chu

## Table of contents

1 Publication Information.....	2
2 Knowledge.....	2
3 Experience.....	4
4 Focus.....	6
5 Look Ahead.....	7

## 1. Publication Information

Copyright ©2004-2008 by Philip Chu All rights reserved.



"The definition of insanity is doing the same thing over and over and expecting different results" - Benjamin Franklin

## 2. Knowledge

### 2.1. Know Thyself

Self-awareness isn't a common trait among self-styled entrepreneurs, but a little self-recognition can save you a lot of trouble.

- I've seen drama queens who had to be the center of attention (regaling employees with personal stories for hours at a time) and seemed to get bored if there wasn't a big fight or

emergency going on. And I've seen control freaks micromanage all aspects of their businesses (to the point of giving employees pop quizzes and approving individual secretarial office supply purchases). Which is fine - passive, unambitious people need jobs, too, and there's plenty of room for small companies specializing in services and small projects. But micromanagement doesn't scale, talented employees don't like to be treated like children, and a large company has to run like a machine, not a series of bar fights. So when these companies succumbed to grand ambitions of growth and innovation, the result wasn't pretty.

Changing your personality is as likely as changing your shoe size. So pick a company that fits.

## **2.2. Know What You'll Do**

The glamorous stories about startups include tales of persistence, passion and nights spent on the office couch. But that isn't always necessary if you just want to be your own boss.

- From experience, I know I can maintain a comfortable lifestyle working on one or two contracts from referrals every year and take it easy between contracts. And I know other self-employed developers who play a lot of golf, spend a lot of quality time with their families, and don't work weekends.

But that's not going to scale into anything big or popular. Be realistic about how much your ambitions will match your own follow-through.

- I know of one outfit that put a customer forum on their web site and didn't monitor it for weeks at a time. After seeing it filled with spam and customer complaints of neglect ("treated like a two-dollar whore" was one phrase), they took it down. Another startup asked me to introduce them to high-level game industry contacts to sell their product, but never got around to that little detail of creating the product.

You should also be aware of what you're willing to do. And what you're not willing to do.

## **2.3. Know What You Want**

Besides knowing who you are, know what you're really trying to achieve. There's the mission statement you put in your business plan to impress VC's, and then there is your real mission. Whether it be getting rich, getting famous, showing your dad you really amounted to something, or getting even with all those people who did you wrong at your last job, you should know, as they say in acting classes, "what's your motivation?"

- I was recruited by the president of a game development company who told me she

wanted to make a "great" game. But she spent more time talking about famous and rich personalities in the industry than about notable game designers or learning about game design. After lots of schmoozing and mediocre development with junior designers who had to clock in like Walmart employees, the result was, mediocre. And no one got rich and famous.

If you know what you can do and what you want to do, maybe your method will match your ambition.

## **2.4. Know What You Know**

It's also important for the company as a whole to stick with what you know. It's hard enough to set up a company, find good people, and raise money. Having to learn an unfamiliar domain and gain connections and credibility in that domain will make success nearly impossible.

- These days, a lot of people want to be Steve Jobs. I was brought into one successful software company that was taken over by a group who had made their fortunes in sales and services, and in short order they managed to run it into the ground. Status meetings consisted of "when can I sell this?" The delivery process amounted to "you're giving me a build today." A process perhaps fit for a burger joint ("you want fries with that build?"), but not for software development.

A significant successful project requires all kinds of people - and chances are, you're not all kinds of people. If you're a technology guy and not a deal-maker, you need to bring in or partner with people who can bring in the deals. If you're a company builder and investment getter and not a technology guy, you probably shouldn't be architecting a new software product. Figure out the critical functions that should be your focus, and do no harm in the other areas.

## **3. Experience**

### **3.1. Learn From It**

It's easy to Monday morning quarterback a failed startup. I've often heard the query "What were they thinking?", but sometimes this is not really a fair question. The reasoning may have seemed sound at the time - certainly if someone provided funding, then the idea must not have been an obviously terrible one.

- The first startup I joined had a pretty reasonable-sounding plan: sell a full-featured 3D graphics content creation system to game developers, based largely on technical

suitability and underpricing the twenty-thousand dollar per unit competing products. Who would've known that Autodesk would soon enter the market with a vastly cheaper product and established presence in the game market? I didn't.

Beware the entrepreneur who only knows (or remembers) success and thinks he has the Midas touch. success with a startup is largely the luck of the draw. Failure, on the other hand, is not a bad thing - you can learn from failure. But some entrepreneurs are repeat offenders.

- I worked for one CTO who exhibited many admirable traits, including loyalty - but to a fault. I was a beneficiary (and much appreciated it), but one of his golden boys returned the favor by alienating the rest of the staff and prompting several senior engineers to leave. (I was promoted quickly in that job) The attrition problem was eventually "solved" by putting the programmer in charge of another group, essentially transferring the problem and giving him a promotion to boot. After the startup was ignominiously absorbed into an acquiring company and quietly extinguished, the CTO chose to start another company with, guess who?

Show me someone who's running a startup, and I'll show you someone with an overabundance of self-confidence and/or someone who's tired of working for other people. Either way, recognition of past mistakes (unless they're made by others) is often not part of the package. But if you're going to go through the school of hard knocks, you might as well learn something.

### **3.2. Use It**

Especially in a startup, you need to take advantage of your staff's existing expertise. Surprisingly, small companies can be as clueless as large ones at recognizing individual assets.

- Despite mentioning repeatedly that I'd worked on military visual simulation projects, the computer graphics company I joined never bothered to consult me when they tried to get into that market. For a big corporation, that's somewhat understandable (although such a corporation should have the resources to set up an employee database to check for in-house talent). For a startup in which budgeting and time-to-market is crucial, there's no excuse - just send email to your staff. How hard is that?
- I spent interminably painful weekly meetings in another startup in which the CEO would raptly listen to the CTO expound on topics he knew nothing about. I'm certain better answers about machine vision and image processing, for example, could have been obtained from the engineer who did his PhD in machine vision. Given that I'd just left a wireless Internet company, it would have made sense to ask me about wireless devices and operating web servers. (They almost tried to invent their own secure communication

with web servers, not realizing there is a standard used for e-commerce).

Even if you stick to your core competence, you're going to run into unfamiliar customers and application domains. Before running blindly into them, query your staff - "Does anyone know anything about this/them?"

## **4. Focus**

### **4.1. Be Cheap**

There used to be a saying that Silicon Valley high-tech startups would decline once they moved into new glamorous digs. That saying referred to high-profile companies like Silicon Graphics, but I've seen this happen even in early-stage startups.

- When I joined a San Francisco wireless internet startup that had just received its Series A financing, they had just moved out of one of the founders' homes and into a moderate-sized office in the Financial District. In three months, they took a long-term lease on half of the top floor (where I had an amazing view of the bay), started purchasing expensive office furniture, and hired three administrative assistants, for a staff totalling no more than twenty. When the next series of financing looked less inevitable, the COO had to clamp down on spending and we ended up with twenty dollar utility tables as desks, to complement our eight-hundred dollar chairs.
- The next startup I joined was based in artsy Venice, CA. So of course the office upgrade involved a complete redesign by an architect involving curved walls and all attached desks and counters had to have matching custom curves. A Fung-Shui consultant then walked around advising how to maximize the good fortune of the space. To no avail, as there were layoffs within a month after the work was finished.

Don't spend money that you don't already have.

### **4.2. But Not Too Cheap**

On the other hand, don't pinch pennies at the expense of getting things done. Starting with people.

- I was offered a bit of sweat equity to take part in a startup but bailed out quickly when I learned everyone else also had a day job. Thus is turned into a fun activity in which participants could play the startup game and brainstorm cool business ideas, instead of an urgent mission. If the founders had paid a full-time contractor they could have had a prototype running in a month instead of blowing at least that much over the course of a year with nothing to show at the end.

And there's no point in investing in people if you're not going to give them the tools to get the job done right, and on time.

- One company president who was ready to hire more engineers to accelerate my project but unwilling to buy more computers. While he wheedled our partners to loan us machines, I had to tussle with the documentation and training groups over the existing workstations.

The saying goes, time is money, but in a startup, money buys you time. Don't waste it.

### **4.3. Start with One Thing**

I've never seen a startup that tried to begin with several products at once succeed. You can be GE later, but start with one project.

- One startup that I consulted for was so optimistic they rented manufacturing space and purchased booth space at CES before they had connected a single wire or written a single line of code. They could have implemented a prototype within a few months just using stock PC hardware, but instead got distracted by different ideas, ranging from robots and educational software to digital content distribution. In the end, they had nothing to show. Another startup that I helped get running also had eyes too big for our stomach. We started developing two rather ambitious computer products and didn't completely follow through on either.

### **4.4. And Finish It**

The easy part of a startup is the starting part. Finishing something is the hardest part. Being the idea guy is fun (consulting is great work if you can get it), but developing, debugging and polishing a product, and then deploying, supporting and maintaining it, is painstaking work that requires tenacity and discipline.

- At the computer graphics startup I helped form, I don't know if we could have sold anything, but certainly we could have had a usable product if I'd bothered to implement file save/export capability. As it was, I left it a demo, and one of the few, and regrettable cases, where I can't point to a finished product. And when we abandoned development, testers who wanted to use our product, couldn't.

So whatever product or service you decide to start out with, get it done before you move onto something else. You need to prove to yourself and to everyone else that you can execute.

## **5. Look Ahead**

## 5.1. Growing Pains

My favorite time to join a startup is when they still have less than ten people - at that time, everyone is still focussed on just getting something going and seeing how it turns out. Once the company headcount reaches fifteen, diverging agendas involving turf issues, managerial rivalries, and career ambitions rear their ugly heads.

- My first job at a startup was chaotic but fun until the headcount reached about twenty - then it seemed everyone had a management title and turf to protect (including me). That was also my first management position, and I was surprised to find that dealing with the engineers in my group was the easy part - dealing with all the other managers was the hard part.

Plan ahead of time how you're going to handle the personality conflicts, communication issues, and different requirements of a larger and more normal workforce.

## 5.2. Keep It Clean

It's all too common to see stories of malfeasance in large, publicly-traded corporations, and the natural reaction is - how could that happen in a professionally-run company? And why would wealthy executives take the risk of bending the rules here and there?

Well, that kind of stuff happens all the time in small companies - it just doesn't get the same headline publicity, unless you watch *The People's Court*. Big companies were once small companies, and big-time executives caught with their hands in the cookie jar used to be small-time players looking for an angle.

- Backdating? Clients routinely take their time putting together contracts and then expect me to backdate them. One former employer contacted me years after I left and asked me to sign and backdate an employment agreement to satisfy a potential acquirer. Stock pumping or fake articles using false identities on the Internet? I found a rave review of a mediocre game that I helped develop at a small game company - it was obvious from the crude writing style that it was written by the company president. Tax evasion? Ummm...no comment.

A lawyer for one of my startups asked us if we wanted to do things "clean" or "dirty" - that shouldn't even be an option! Don't get started with bad habits - if you're considering something that would get you in trouble in a larger company, just don't do it.

And it's not just a matter of staying out of jail. A dicey reputation is a lot easier to pick up than to get rid of. Whenever I have a prospective client or vendor, I ask around - sometimes the answer is "run the other way". And the way you conduct business will become ingrained

in your company - that can come back to haunt you.

- One marketing guy reminded me of the pathological liar character on Saturday Night Live. He prevaricated like other people breath. He told me he was an RAF jet fighter pilot. He informed another coworker that he held a patent for a particular printer technology. He said he needed time off because his wife was pregnant (not). And then he skipped to a direct competitor.

No one wants to deal with a company that can't even trust its own employees.

### **5.3. Have an Exit Strategy**

As in war and casinos, you should have an exit strategy. Are you aiming to cash out via an IPO or acquisition by another company? Do you plan to run this company indefinitely? What are you going to do if the current product or strategy doesn't work out or if you fail to bring in enough revenue or financing? Change strategy? Declare bankruptcy? You should have a Plan B whether it involves cashing out on top or cutting your losses.