

Why Smart People Have Bad Ideas

paulgraham.com · Paul Graham · 2005-04 · [source](#)

| | |

|

April 2005

This summer, as an experiment, some friends and I are giving seed funding to a bunch of new startups. It's an experiment because we're prepared to fund younger founders than most investors would. That's why we're doing it during the summer—so even college students can participate.

We know from Google and Yahoo that grad students can start successful startups. And we know from experience that some undergrads are as capable as most grad students. The accepted age for startup founders has been creeping downward. We're trying to find the lower bound.

The deadline has now passed, and we're sifting through 227 applications. We expected to divide them into two categories, promising and unpromising. But we soon saw we needed a third: promising people with unpromising ideas.

[1]

The Artix Phase

We should have expected this. It's very common for a group of founders to go through one lame idea before realizing that a startup has to make something people will pay for. In fact, we ourselves did.

Viaweb wasn't the first startup Robert Morris and I started. In January 1995, we and a couple friends started a company called

Artix. The plan was to put art galleries on the Web. In retrospect, I wonder how we could have wasted our time on anything so stupid. Galleries are not especially excited about being on the Web even now, ten years later. They don't want to have their stock visible to any random visitor, like an antique store.

[2]

Besides which, art dealers are the most technophobic people on earth. They didn't become art dealers after a difficult choice between that and a career in the hard sciences. Most of them had never seen the Web before we came to tell them why they should be on it. Some didn't even have computers. It doesn't do justice to the situation to describe it as a hard sell; we soon sank to building sites for free, and it was hard to convince galleries even to do that.

Gradually it dawned on us that instead of trying to make Web sites for people who didn't want them, we could make sites for people who did. In fact, software that would let people who wanted sites make their own. So we ditched Artix and started a new company, Viaweb, to make software for building online stores. That one succeeded.

We're in good company here. Microsoft was not the first company Paul Allen and Bill Gates started either. The first was called Traf-o-data. It does not seem to have done as well as Micro-soft.

In Robert's defense, he was skeptical about Artix. I dragged him into it.

[3]

But there were moments when he was optimistic. And if we, who were 29 and 30 at the time, could get excited about such a thoroughly boneheaded idea, we should not be surprised that hackers aged 21 or 22 are pitching us ideas with little hope of making money.

The Still Life Effect

Why does this happen? Why do good hackers have bad business ideas?

Let's look at our case. One reason we had such a lame idea was that it was the first thing we thought of. I was in New York trying to be a starving artist at the time (the starving part is actually quite easy), so I was haunting galleries anyway. When I learned about the Web, it seemed natural to mix the two. Make Web sites for galleries—that's the ticket!

If you're going to spend years working on something, you'd think it might be wise to spend at least a couple days considering different ideas, instead of going with the first that comes into your head. You'd think. But people don't. In fact, this is a constant problem when you're painting still lifes. You plonk down a bunch of stuff on a table, and maybe spend five or ten minutes rearranging it to look interesting. But you're so impatient to get started painting that ten minutes of rearranging feels very long. So you start painting. Three days later, having spent twenty hours staring at it, you're kicking yourself for having set up such an awkward and boring composition, but by then it's too late.

Part of the problem is that big projects tend to grow out of small ones. You set up a still life to make a quick sketch when you have a spare hour, and days later you're still working on it. I once spent a month painting three versions of a still life I set up in about four minutes. At each point (a day, a week, a month) I thought I'd already put in so much time that it was too late to change.

So the biggest cause of bad ideas is the still life effect: you come up with a random idea, plunge into it, and then at each point (a day, a week, a month) feel you've put so much time into it that this must be the idea.

How do we fix that? I don't think we should discard plunging. Plunging into an idea is a good thing. The solution is at the other end: to realize that having invested time in something doesn't make it good.

This is clearest in the case of names. Viaweb was originally called Webgen, but we discovered someone else had a product called that. We were so attached to our name that we offered him 5% of the company if he'd let us have it. But he wouldn't, so

we had to think of another.

[4]

The best we could do was Viaweb, which we disliked at first. It was like having a new mother. But within three days we loved it, and Webgen sounded lame and old-fashioned.

If it's hard to change something so simple as a name, imagine how hard it is to garbage-collect an idea. A name only has one point of attachment into your head. An idea for a company gets woven into your thoughts. So you must consciously discount for that. Plunge in, by all means, but remember later to look at your idea in the harsh light of morning and ask: is this something people will pay for? Is this, of all the things we could make, the thing people will pay most for?

Muck

The second mistake we made with Artix is also very common. Putting galleries on the Web seemed cool.

One of the most valuable things my father taught me is an old Yorkshire saying: where there's muck, there's brass. Meaning that unpleasant work pays. And more to the point here, vice versa. Work people like doesn't pay well, for reasons of supply and demand. The most extreme case is developing programming languages, which doesn't pay at all, because people like it so much they do it for free.

When we started Artix, I was still ambivalent about business. I wanted to keep one foot in the art world. Big, big, mistake. Going into business is like a hang-glider launch: you'd better do it wholeheartedly, or not at all. The purpose of a company, and a startup especially, is to make money. You can't have divided loyalties.

Which is not to say that you have to do the most disgusting sort of work, like spamming, or starting a company whose only purpose is patent litigation. What I mean is, if you're starting a company that will do something cool, the aim had better be to make money

and maybe be cool, not to be cool and maybe make money.

It's hard enough to make money that you can't do it by accident. Unless it's your first priority, it's unlikely to happen at all.

Hyenas

When I probe our motives with Artix, I see a third mistake: timidity. If you'd proposed at the time that we go into the e-commerce business, we'd have found the idea terrifying. Surely a field like that would be dominated by fearsome startups with five million dollars of VC money each. Whereas we felt pretty sure that we could hold our own in the slightly less competitive business of generating Web sites for art galleries.

We erred ridiculously far on the side of safety. As it turns out, VC-backed startups are not that fearsome. They're too busy trying to spend all that money to get software written. In 1995, the e-commerce business was very competitive as measured in press releases, but not as measured in software. And really it never was. The big fish like Open Market (rest their souls) were just consulting companies pretending to be product companies [5], and the offerings at our end of the market were a couple hundred lines of Perl scripts. Or could have been implemented as a couple hundred lines of Perl; in fact they were probably tens of thousands of lines of C++ or Java. Once we actually took the plunge into e-commerce, it turned out to be surprisingly easy to compete.

So why were we afraid? We felt we were good at programming, but we lacked confidence in our ability to do a mysterious, undifferentiated thing we called "business." In fact there is no such thing as "business." There's selling, promotion, figuring out what people want, deciding how much to charge, customer support, paying your bills, getting customers to pay you, getting incorporated, raising money, and so on. And the combination is not as hard as it seems, because some tasks (like raising money and getting incorporated) are an $O(1)$ pain in the ass, whether you're big or small, and others (like selling and promotion) depend more on energy and imagination

than any kind of special training.

Artix was like a hyena, content to survive on carrion because we were afraid of the lions. Except the lions turned out not to have any teeth, and the business of putting galleries online barely qualified as carrion.

A Familiar Problem

Sum up all these sources of error, and it's no wonder we had such a bad idea for a company. We did the first thing we thought of; we were ambivalent about being in business at all; and we deliberately chose an impoverished market to avoid competition.

Looking at the applications for the Summer Founders Program, I see signs of all three. But the first is by far the biggest problem. Most of the groups applying have not stopped to ask: of all the things we could do, is this the one with the best chance of making money?

If they'd already been through their Artix phase, they'd have learned to ask that. After the reception we got from art dealers, we were ready to. This time, we thought, let's make something people want.

Reading the Wall Street Journal for a week should give anyone ideas for two or three new startups. The articles are full of descriptions of problems that need to be solved. But most of the applicants don't seem to have looked far for ideas.

We expected the most common proposal to be for multiplayer games. We were not far off: this was the second most common. The most common was some combination of a blog, a calendar, a dating site, and Friendster. Maybe there is some new killer app to be discovered here, but it seems perverse to go poking around in this fog when there are valuable, unsolved problems lying about in the open for anyone to see. Why did no one propose a new scheme for micropayments? An ambitious project, perhaps, but I can't believe we've considered every alternative. And newspapers and magazines are (literally) dying for a solution.

Why did so few applicants really think about what customers want? I think the problem with many, as with people in their early twenties generally, is that they've been trained their whole lives to jump through predefined hoops. They've spent 15-20 years solving problems other people have set for them. And how much time deciding what problems would be good to solve? Two or three course projects? They're good at solving problems, but bad at choosing them.

But that, I'm convinced, is just the effect of training. Or more precisely, the effect of grading. To make grading efficient, everyone has to solve the same problem, and that means it has to be decided in advance. It would be great if schools taught students how to choose problems as well as how to solve them, but I don't know how you'd run such a class in practice.

Copper and Tin

The good news is, choosing problems is something that can be learned. I know that from experience. Hackers can learn to make things customers want.

[6]

This is a controversial view. One expert on "entrepreneurship" told me that any startup had to include business people, because only they could focus on what customers wanted. I'll probably alienate this guy forever by quoting him, but I have to risk it, because his email was such a perfect example of this view:

80% of MIT spinoffs succeed provided they have at least one management person in the team at the start. The business person represents the "voice of the customer" and that's what keeps the engineers and product development on track. This is, in my opinion, a crock. Hackers are perfectly capable of hearing the voice of the customer without a business person to amplify the signal for them. Larry Page and Sergey Brin were grad students in computer science, which presumably makes them "engineers." Do you suppose Google is only good because they had some business guy whispering in their ears what customers wanted? It seems to me the business guys who did the most for Google were the ones who obligingly flew Altavista into a hillside just as Google was getting

started.

The hard part about figuring out what customers want is figuring out that you need to figure it out. But that's something you can learn quickly. It's like seeing the other interpretation of an ambiguous picture. As soon as someone tells you there's a rabbit as well as a duck, it's hard not to see it.

And compared to the sort of problems hackers are used to solving, giving customers what they want is easy. Anyone who can write an optimizing compiler can design a UI that doesn't confuse users, once they choose to focus on that problem. And once you apply that kind of brain power to petty but profitable questions, you can create wealth very rapidly.

That's the essence of a startup: having brilliant people do work that's beneath them. Big companies try to hire the right person for the job. Startups win because they don't—because they take people so smart that they would in a big company be doing "research," and set them to work instead on problems of the most immediate and mundane sort. Think Einstein designing refrigerators.

[7]

If you want to learn what people want, read Dale Carnegie's *How to Win Friends and Influence People*.

[8]

When a friend recommended this book, I couldn't believe he was serious. But he insisted it was good, so I read it, and he was right. It deals with the most difficult problem in human experience: how to see things from other people's point of view, instead of thinking only of yourself.

Most smart people don't do that very well. But adding this ability to raw brainpower is like adding tin to copper. The result is bronze, which is so much harder that it seems a different metal.

A hacker who has learned what to make, and not just how to make, is extraordinarily powerful. And not just at making money: look what a small group of volunteers has achieved with Firefox.

Doing an Artix teaches you to make something people want in the

same way that not drinking anything would teach you how much you depend on water. But it would be more convenient for all involved if the Summer Founders didn't learn this on our dime—if they could skip the Artix phase and go right on to make something customers wanted. That, I think, is going to be the real experiment this summer. How long will it take them to grasp this?

We decided

we ought to have T-Shirts for the SFP, and we'd been thinking about what to print on the back. Till now we'd been planning to use

If you can read this, I should be working, but now we've decided it's going to be Make something people want.

Notes

[1]

SFP applicants: please don't assume that not being accepted means we think your idea is bad. Because we want to keep the number of startups small this first summer, we're going to have to turn down some good proposals too.

[2]

Dealers try to give each customer the impression that the stuff they're showing him is something special that only a few people have seen, when in fact it may have been sitting in their racks for years while they tried to unload it on buyer after buyer.

[3]

On the other hand, he was skeptical about Viaweb too. I have a precise measure of that, because at one point in the first couple months we made a bet: if he ever made a million dollars out of Viaweb, he'd get his ear pierced. We didn't let him off, either.

[4]

I wrote a program to generate all the combinations of "Web" plus a three letter word. I learned from this that most three letter words are bad: Webpig, Webdog, Webfat, Webzit, Webfug. But one of them was Webvia; I swapped them to make Viaweb.

[5]

It's much easier to sell services than a product, just as it's easier to make a living playing at weddings than by selling recordings. But the margins are greater on products. So during the Bubble a lot of companies used consulting to generate revenues they could attribute to the sale of products, because it made a better story for an IPO.

[6]

Trevor Blackwell presents the following recipe for a startup: "Watch people who have money to spend, see what they're wasting their time on, cook up a solution, and try selling it to them. It's surprising how small a problem can be and still provide a profitable market for a solution."

[7]

You need to offer especially large rewards to get great people to do tedious work. That's why startups always pay equity rather than just salary.

[8]

Buy an old copy from the 1940s or 50s instead of the current edition, which has been rewritten to suit present fashions. The original edition contained a few unPC ideas, but it's always better to read an original book, bearing in mind that it's a book from a past era, than to read a new version sanitized for your protection.

Thanks to Bill Birch, Trevor Blackwell, Jessica Livingston, and Robert Morris for reading drafts of this.

|

|