Six Principles for Making New Things

paulgraham.com · Paul Graham · 2008-02 · source

| | |

|

February 2008

The fiery reaction to the release of Arc had
an unexpected consequence: it made me realize I had a design
philosophy. The main complaint of the more articulate critics was
that Arc seemed so flimsy. After years of working on it, all I had
to show for myself were a few thousand lines of macros? Why hadn't
I worked on more substantial problems?

As I was mulling over these remarks it struck me how familiar they
seemed. This was exactly the kind of thing people said at first
about Viaweb, and Y Combinator, and most of my essays.

When we launched Viaweb, it seemed laughable to VCs and e-commerce
"experts." We were just a couple guys in an apartment,
which did not seem cool in 1995 the way it does now. And the thing
we'd built, as far as they could tell, wasn't even software.
Software, to them, equalled big, honking Windows apps. Since Viaweb
was the first web-based app
they'd seen, it seemed to be nothing
more than a website. They were even more contemptuous when they
discovered that Viaweb didn't process credit card transactions (we
didn't for the whole first year). Transaction processing seemed
to them what e-commerce was all about. It sounded serious and
difficult.

And yet, mysteriously, Viaweb ended up crushing all its competitors.

The initial reaction to

Y Combinator was almost identical. It seemed laughably lightweight. Startup funding meant series A rounds: millions of dollars given to a small number of startups founded by people with established credentials after months of serious, businesslike meetings, on terms described in a document a foot thick. Y Combinator seemed inconsequential. It's too early to say yet whether Y Combinator will turn out like Viaweb, but judging from the number of imitations, a lot of people seem to think we're on to something.

I can't measure whether my essays are successful, except in page views, but the reaction to them is at least different from when I started. At first the default reaction of the Slashdot trolls was (translated into articulate terms): "Who is this guy and what authority does he have to write about these topics? I haven't read the essay, but there's no way anything so short and written in such an informal style could have anything useful to say about such and such topic, when people with degrees in the subject have already written many thick books about it." Now there's a new generation of trolls on a new generation of sites, but they have at least started to omit the initial "Who is this guy?"

Now people are saying the same things about Arc that they said at first about Viaweb and Y Combinator and most of my essays. Why the pattern? The answer, I realized, is that my m.o. for all four has been the same.

Here it is: I like to find (a) simple solutions (b) to overlooked problems (c) that actually need to be solved, and (d) deliver them as informally as possible, (e) starting with a very crude version 1, then (f) iterating rapidly.

When I first laid out these principles explicitly, I noticed something striking: this is practically a recipe for generating a contemptuous initial reaction. Though simple solutions are better, they don't seem as impressive as complex ones. Overlooked problems are by definition problems that most people think don't matter. Delivering solutions in an informal way means that instead of judging something by the way it's presented, people have to actually understand it, which is more work. And starting with a crude version 1 means your

initial effort is always small and incomplete.

I'd noticed, of course, that people never seemed to grasp new ideas at first. I thought it was just because most people were stupid. Now I see there's more to it than that. Like a contrarian investment fund, someone following this strategy will almost always be doing things that seem wrong to the average person.

As with contrarian investment strategies, that's exactly the point. This technique is successful (in the long term) because it gives you all the advantages other people forgo by trying to seem legit. If you work on overlooked problems, you're more likely to discover new things, because you have less competition. If you deliver solutions informally, you (a) save all the effort you would have had to expend to make them look impressive, and (b) avoid the danger of fooling yourself as well as your audience. And if you release a crude version 1 then iterate, your solution can benefit from the imagination of nature, which, as Feynman pointed out, is more powerful than your own.

In the case of Viaweb, the simple solution was to make the software run on the server. The overlooked problem was to generate web sites automatically; in 1995, online stores were all made by hand by human designers, but we knew this wouldn't scale. The part that actually mattered was graphic design, not transaction processing. The informal delivery mechanism was me, showing up in jeans and a t-shirt at some retailer's office. And the crude version 1 was, if I remember correctly, less than 10,000 lines of code when we launched.

The power of this technique extends beyond startups and programming languages and essays. It probably extends to any kind of creative work. Certainly it can be used in painting: this is exactly what Cezanne and Klee did.

At Y Combinator we bet money on it, in the sense that we encourage the startups we fund to work this way. There are always new ideas right under your nose. So look for simple things that other people have overlooked—things people will later claim were "obvious"—especially when they've been led astray by obsolete

conventions,
or by trying to do things that are superficially impressive. Figure out what the real problem is, and make sure you solve that. Don't worry about trying to look corporate; the product is what wins in the long term. And launch as soon as you can, so you start learning from users what you should have been making.

Reddit is a classic example of this approach. When Reddit first launched, it seemed like there was nothing to it. To the graphically unsophisticated its deliberately minimal design seemed like no design at all. But Reddit solved the real problem, which was to tell people what was new and otherwise stay out of the way. As a result it became massively successful. Now that conventional ideas have caught up with it, it seems obvious. People look at Reddit and think the founders were lucky. Like all such things, it was harder than it looked. The Reddits pushed so hard against the current that they reversed it; now it looks like they're merely floating downstream.

So when you look at something like Reddit and think "I wish I could think of an idea like that," remember: ideas like that are all around you. But you ignore them because they look wrong.

|
|