Home
Essays
H&P
Books
YC
Arc
Bel
Lisp
Spam
Responses
FAQs
RAQs
Quotes
RSS
Bio
Twitter

What I Did this Summer

October 2005

The first Summer Founders Program has just finished. We were surprised how well it went. Overall only about 10% of startups succeed, but if I had to guess now, I'd predict three or four of the eight startups we funded will make it.

Of the startups that needed further funding, I believe all have either closed a round or are likely to soon. Two have already turned down (lowball) acquisition offers.

We would have been happy if just one of the eight seemed promising by the end of the summer. What's going on? Did some kind of anomaly make this summer's applicants especially good? We worry about that, but we can't think of one. We'll find out this winter.

The whole summer was full of surprises. The best was that the hypothesis we were testing seems to be correct. Young hackers can start viable companies. This is good news for two reasons: (a) it's an encouraging thought, and (b) it means that Y Combinator, which is predicated on the idea, is not hosed.

Age

More precisely, the hypothesis was that success in a startup depends mainly on how smart and energetic you are, and much less on how old you are or how much business experience you have. The results so far bear this out. The 2005 summer founders ranged in age from 18 to 28 (average 23), and there is no correlation between their ages and how well they're doing.

This should not really be surprising. Bill Gates and Michael Dell were both 19 when they started the companies that made them

famous. Young founders are not a new phenomenon: the trend began as soon as computers got cheap enough for college kids to afford them.

Another of our hypotheses was that you can start a startup on less money than most people think. Other investors were surprised to hear the most we gave any group was $20,000. But we knew it was possible to start on that little because we started Viaweb on $10,000.

And so it proved this summer. Three months' funding is enough to get into second gear. We had a demo day for potential investors ten weeks in, and seven of the eight groups had a prototype ready by that time. One, Reddit, had already launched, and were able to give a demo of their live site.

A researcher who studied the SFP startups said the one thing they had in common was that they all worked ridiculously hard. People this age are commonly seen as lazy. I think in some cases it's not so much that they lack the appetite for work, but that the work they're offered is unappetizing.

The experience of the SFP suggests that if you let motivated people do real work, they work hard, whatever their age. As one of the founders said "I'd read that starting a startup consumed your life, but I had no idea what that meant until I did it."

I'd feel guilty if I were a boss making people work this hard. But we're not these people's bosses. They're working on their own projects. And what makes them work is not us but their competitors. Like good athletes, they don't work hard because the coach yells at them, but because they want to win.

We have less power than bosses, and yet the founders work harder than employees. It seems like a win for everyone. The only catch is that we get on average only about 5-7% of the upside, while an employer gets nearly all of it. (We're counting on it being 5-7% of a much larger number.)

As well as working hard, the groups all turned out to be extraordinarily responsible. I can't think of a time when one failed to do something they'd promised to, even by being late for an appointment. This is another lesson the world has yet to learn. One of the founders discovered that the hardest part of arranging a meeting with executives at a big cell phone carrier was getting a rental company to rent him a car, because he was too young.

I think the problem here is much the same as with the apparent laziness of people this age. They seem lazy because the work they're given is pointless, and they act irresponsible because they're not given any power. Some of them, anyway. We only have a sample size of about twenty, but it seems so far that if you let people in their early twenties be their own bosses, they rise to the occasion.

Morale

The summer founders were as a rule very idealistic. They also wanted very much to get rich. These qualities might seem incompatible, but they're not. These guys want to get rich, but they want to do it by changing the world. They wouldn't (well, seven of the eight groups wouldn't) be interested in making money by speculating in stocks. They want to make something people use.

I think this makes them more effective as founders. As hard as people will work for money, they'll work harder for a cause. And since success in a startup depends so much on motivation, the paradoxical result is that the people likely to make the most money are those who aren't in it just for the money.

The founders of Kiko, for example, are working on an Ajax calendar. They want to get rich, but they pay more attention to design than they would if that were their only motivation. You can tell just by looking at it.

I never considered it till this summer, but this might be another reason startups run by hackers tend to do better than those run by MBAs. Perhaps it's not just that hackers understand technology better, but that they're driven by more powerful motivations. Microsoft, as I've said before, is a dangerously misleading example. Their mean corporate culture only works for monopolies. Google is a better model.

Considering that the summer founders are the sharks in this ocean, we were surprised how frightened most of them were of competitors. But now that I think of it, we were just as frightened when we started Viaweb. For the first year, our initial reaction to news of a competitor was always: we're doomed. Just as a hypochondriac magnifies his symptoms till he's convinced he has some terrible disease, when you're not used to competitors you magnify them into monsters.

Here's a handy rule for startups: competitors are rarely as dangerous as they seem. Most will self-destruct before you can destroy them. And it certainly doesn't matter how many of them there are, any more than it matters to the winner of a marathon how many runners are behind him.

"It's a crowded market," I remember one founder saying worriedly.

"Are you the current leader?" I asked.

"Yes."

"Is anyone able to develop software faster than you?"

"Probably not."

"Well, if you're ahead now, and you're the fastest, then you'll stay ahead. What difference does it make how many others there are?"

Another group was worried when they realized they had to rewrite their software from scratch. I told them it would be a bad sign if they didn't. The main function of your initial version is to be rewritten.

That's why we advise groups to ignore issues like scalability, internationalization, and heavy-duty security at first. [1] I can imagine an advocate of "best practices" saying these ought to be considered from the start. And he'd be right, except that they interfere with the primary function of software in a startup: to be a vehicle for experimenting with its own design. Having to retrofit internationalization or scalability is a pain, certainly. The only bigger pain is not needing to, because your initial version was too big and rigid to evolve into something users wanted.

I suspect this is another reason startups beat big companies. Startups can be irresponsible and release version 1s that are light enough to evolve. In big companies, all the pressure is in the direction of over-engineering.

What Got Learned

One thing we were curious about this summer was where these groups would need help. That turned out to vary a lot. Some we helped with technical advice-- for example, about how to set up an application to run on multiple servers. Most we helped with strategy questions, like what to patent, and what to charge for and what to give away. Nearly all wanted advice about dealing with future investors: how much money should they take and what kind of terms should they expect?

However, all the groups quickly learned how to deal with stuff like patents and investors. These problems aren't intrinsically difficult, just unfamiliar.

It was surprising-- slightly frightening even-- how fast they learned. The weekend before the demo day for investors, we had a practice session where all the groups gave their presentations. They were all terrible. We tried to explain how to make them better, but we didn't have much hope. So on demo day I told the assembled angels and VCs that these guys were hackers, not MBAs, and so while their software was good, we should not expect slick presentations from them.

The groups then proceeded to give fabulously slick presentations. Gone were the mumbling recitations of lists of features. It was as if they'd spent the past week at acting school. I still don't know how they did it.

Perhaps watching each others' presentations helped them see what they'd been doing wrong. Just as happens in college, the summer founders learned a lot from one another-- maybe more than they learned from us. A lot of the problems they face are the same, from dealing with investors to hacking Javascript.

I don't want to give the impression there were no problems this summer. A lot went wrong, as usually happens with startups. One

group got an "exploding term-sheet" from some VCs. Pretty much all the groups who had dealings with big companies found that big companies do everything infinitely slowly. (This is to be expected. If big companies weren't incapable, there would be no room for startups to exist.) And of course there were the usual nightmares associated with servers.

In short, the disasters this summer were just the usual childhood diseases. Some of this summer's eight startups will probably die eventually; it would be extraordinary if all eight succeeded. But what kills them will not be dramatic, external threats, but a mundane, internal one: not getting enough done.

So far, though, the news is all good. In fact, we were surprised how much fun the summer was for us. The main reason was how much we liked the founders. They're so earnest and hard-working. They seem to like us too. And this illustrates another advantage of investing over hiring: our relationship with them is way better than it would be between a boss and an employee. Y Combinator ends up being more like an older brother than a parent.

I was surprised how much time I spent making introductions. Fortunately I discovered that when a startup needed to talk to someone, I could usually get to the right person by at most one hop. I remember wondering, how did my friends get to be so eminent? and a second later realizing: shit, I'm forty.

Another surprise was that the three-month batch format, which we were forced into by the constraints of the summer, turned out to be an advantage. When we started Y Combinator, we planned to invest the way other venture firms do: as proposals came in, we'd evaluate them and decide yes or no. The SFP was just an experiment to get things started. But it worked so well that we plan to do all our investing this way, one cycle in the summer and one in winter. It's more efficient for us, and better for the startups too.

Several groups said our weekly dinners saved them from a common problem afflicting startups: working so hard that one has no social life. (I remember that part all too well.) This way, they were guaranteed a social event at least once a week.

Independence

I've heard Y Combinator described as an "incubator." Actually we're the opposite: incubators exert more control than ordinary VCs, and we make a point of exerting less. Among other things, incubators usually make you work in their office-- that's where the word "incubator" comes from. That seems the wrong model. If investors get too involved, they smother one of the most powerful forces in a startup: the feeling that it's your own company.

Incubators were conspicuous failures during the Bubble. There's still debate about whether this was because of the Bubble, or because they're a bad idea. My vote is they're a bad idea. I think

they fail because they select for the wrong people. When we were starting a startup, we would never have taken funding from an "incubator." We can find office space, thanks; just give us the money. And people with that attitude are the ones likely to succeed in startups.

Indeed, one quality all the founders shared this summer was a spirit of independence. I've been wondering about that. Are some people just a lot more independent than others, or would everyone be this way if they were allowed to?

As with most nature/nurture questions, the answer is probably: some of each. But my main conclusion from the summer is that there's more environment in the mix than most people realize. I could see that from how the founders' attitudes *changed* during the summer. Most were emerging from twenty or so years of being told what to do. They seemed a little surprised at having total freedom. But they grew into it really quickly; some of these guys now seem about four inches taller (metaphorically) than they did at the beginning of the summer.

When we asked the summer founders what surprised them most about starting a company, one said "the most shocking thing is that it worked."

It will take more experience to know for sure, but my guess is that a lot of hackers could do this-- that if you put people in a position of independence, they develop the qualities they need. Throw them off a cliff, and most will find on the way down that they have wings.

The reason this is news to anyone is that the same forces work in the other direction too. Most hackers are employees, and this molds you into someone to whom starting a startup seems impossible as surely as starting a startup molds you into someone who can handle it.

If I'm right, "hacker" will mean something different in twenty years than it does now. Increasingly it will mean the people who run the company. Y Combinator is just accelerating a process that would have happened anyway. Power is shifting from the people who deal with money to the people who create technology, and if our experience this summer is any guide, this will be a good thing.

Notes

[1] By heavy-duty security I mean efforts to protect against truly determined attackers.

The image shows us, the 2005 summer founders, and Smartleaf co-founders Mark Nitzberg and Olin Shivers at the 30-foot table Kate Courteau designed for us. Photo by Alex Lewin.

Thanks to Sarah Harlin, Steve Huffman, Jessica Livingston, Zak

Stone, and Aaron Swartz for reading drafts of this.

- Romanian Translation
- Japanese Translation
