

Before the Startup

paulgraham.com · Paul Graham · 2014-10 · [source](#)

|

October 2014

(This essay is derived from a guest lecture in Sam Altman's startup class at Stanford. It's intended for college students, but much of it is applicable to potential founders at other ages.)

One of the advantages of having kids is that when you have to give advice, you can ask yourself "what would I tell my own kids?" My kids are little, but I can imagine what I'd tell them about startups if they were in college, and that's what I'm going to tell you.

Startups are very counterintuitive. I'm not sure why. Maybe it's just because knowledge about them hasn't permeated our culture yet. But whatever the reason, starting a startup is a task where you can't always trust your instincts.

It's like skiing in that way. When you first try skiing and you want to slow down, your instinct is to lean back. But if you lean back on skis you fly down the hill out of control. So part of learning to ski is learning to suppress that impulse. Eventually you get new habits, but at first it takes a conscious effort. At first there's a list of things you're trying to remember as you start down the hill.

Startups are as unnatural as skiing, so there's a similar list for startups. Here I'm going to give you the first part of it — the things to remember if you want to prepare yourself to start a startup.

Counterintuitive

The first item on it is the fact I already mentioned: that startups are so weird that if you trust your instincts, you'll make a lot

of mistakes. If you know nothing more than this, you may at least pause before making them.

When I was running Y Combinator I used to joke that our function was to tell founders things they would ignore. It's really true. Batch after batch, the YC partners warn founders about mistakes they're about to make, and the founders ignore them, and then come back a year later and say "I wish we'd listened."

Why do the founders ignore the partners' advice? Well, that's the thing about counterintuitive ideas: they contradict your intuitions. They seem wrong. So of course your first impulse is to disregard them. And in fact my joking description is not merely the curse of Y Combinator but part of its *raison d'être*. If founders' instincts already gave them the right answers, they wouldn't need us. You only need other people to give you advice that surprises you. That's why there are a lot of ski instructors and not many running instructors.

[1]

You can, however, trust your instincts about people. And in fact one of the most common mistakes young founders make is not to do that enough. They get involved with people who seem impressive, but about whom they feel some misgivings personally. Later when things blow up they say "I knew there was something off about him, but I ignored it because he seemed so impressive."

If you're thinking about getting involved with someone — as a cofounder, an employee, an investor, or an acquirer — and you have misgivings about them, trust your gut. If someone seems slippery, or bogus, or a jerk, don't ignore it.

This is one case where it pays to be self-indulgent. Work with people you genuinely like, and you've known long enough to be sure.

Expertise

The second counterintuitive point is that it's not that important to know a lot about startups. The way to succeed in a startup is not to be an expert on startups, but to be an expert on your users

and the problem you're solving for them.

Mark Zuckerberg didn't succeed because he was an expert on startups. He succeeded despite being a complete noob at startups, because he understood his users really well.

If you don't know anything about, say, how to raise an angel round, don't feel bad on that account. That sort of thing you can learn when you need to, and forget after you've done it.

In fact, I worry it's not merely unnecessary to learn in great detail about the mechanics of startups, but possibly somewhat dangerous. If I met an undergrad who knew all about convertible notes and employee agreements and (God forbid) class FF stock, I wouldn't think "here is someone who is way ahead of their peers." It would set off alarms. Because another of the characteristic mistakes of young founders is to go through the motions of starting a startup. They make up some plausible-sounding idea, raise money at a good valuation, rent a cool office, hire a bunch of people. From the outside that seems like what startups do. But the next step after rent a cool office and hire a bunch of people is: gradually realize how completely fucked they are, because while imitating all the outward forms of a startup they have neglected the one thing that's actually essential: making something people want.

Game

We saw this happen so often that we made up a name for it: playing house. Eventually I realized why it was happening. The reason young founders go through the motions of starting a startup is because that's what they've been trained to do for their whole lives up to that point. Think about what you have to do to get into college, for example. Extracurricular activities, check. Even in college classes most of the work is as artificial as running laps.

I'm not attacking the educational system for being this way. There will always be a certain amount of fakeness in the work you do when you're being taught something, and if you measure their performance it's inevitable that people will exploit the difference to the point where much of what you're measuring is artifacts of the fakeness.

I confess I did it myself in college. I found that in a lot of classes there might only be 20 or 30 ideas that were the right shape to make good exam questions. The way I studied for exams in these classes was not (except incidentally) to master the material taught in the class, but to make a list of potential exam questions and work out the answers in advance. When I walked into the final, the main thing I'd be feeling was curiosity about which of my questions would turn up on the exam. It was like a game.

It's not surprising that after being trained for their whole lives to play such games, young founders' first impulse on starting a startup is to try to figure out the tricks for winning at this new game. Since fundraising appears to be the measure of success for startups (another classic noob mistake), they always want to know what the tricks are for convincing investors. We tell them the best way to convince investors is to make a startup that's actually doing well, meaning growing fast, and then simply tell investors so. Then they want to know what the tricks are for growing fast. And we have to tell them the best way to do that is simply to make something people want.

So many of the conversations YC partners have with young founders begin with the founder asking "How do we..." and the partner replying "Just..."

Why do the founders always make things so complicated? The reason, I realized, is that they're looking for the trick.

So this is the third counterintuitive thing to remember about startups: starting a startup is where gaming the system stops working. Gaming the system may continue to work if you go to work for a big company. Depending on how broken the company is, you can succeed by sucking up to the right people, giving the impression of productivity, and so on.

[2]

But that doesn't work with startups.

There is no boss to trick, only users, and all users care about is whether your product does what they want. Startups are as impersonal as physics. You have to make something people want, and you prosper only to the extent you do.

The dangerous thing is, faking does work to some degree on investors. If you're super good at sounding like you know what you're talking about, you can fool investors for at least one and perhaps even two rounds of funding. But it's not in your interest to. The company is ultimately doomed. All you're doing is wasting your own time riding it down.

So stop looking for the trick. There are tricks in startups, as there are in any domain, but they are an order of magnitude less important than solving the real problem. A founder who knows nothing about fundraising but has made something users love will have an easier time raising money than one who knows every trick in the book but has a flat usage graph. And more importantly, the founder who has made something users love is the one who will go on to succeed after raising the money.

Though in a sense it's bad news in that you're deprived of one of your most powerful weapons, I think it's exciting that gaming the system stops working when you start a startup. It's exciting that there even exist parts of the world where you win by doing good work. Imagine how depressing the world would be if it were all like school and big companies, where you either have to spend a lot of time on bullshit things or lose to people who do.

[3]

I would

have been delighted if I'd realized in college that there were parts of the real world where gaming the system mattered less than others, and a few where it hardly mattered at all. But there are, and this variation is one of the most important things to consider when you're thinking about your future. How do you win in each type of work, and what would you like to win by doing?

[4]

All-Consuming

That brings us to our fourth counterintuitive point: startups are all-consuming. If you start a startup, it will take over your life to a degree you cannot imagine. And if your startup succeeds, it will take over your life for a long time: for several years at the

very least, maybe for a decade, maybe for the rest of your working life. So there is a real opportunity cost here.

Larry Page may seem to have an enviable life, but there are aspects of it that are unenviable. Basically at 25 he started running as fast as he could and it must seem to him that he hasn't stopped to catch his breath since. Every day new shit happens in the Google empire that only the CEO can deal with, and he, as CEO, has to deal with it. If he goes on vacation for even a week, a whole week's backlog of shit accumulates. And he has to bear this uncomplainingly, partly because as the company's daddy he can never show fear or weakness, and partly because billionaires get less than zero sympathy if they talk about having difficult lives. Which has the strange side effect that the difficulty of being a successful startup founder is concealed from almost everyone except those who've done it.

Y Combinator has now funded several companies that can be called big successes, and in every single case the founders say the same thing. It never gets any easier. The nature of the problems change. You're worrying about construction delays at your London office instead of the broken air conditioner in your studio apartment. But the total volume of worry never decreases; if anything it increases.

Starting a successful startup is similar to having kids in that it's like a button you push that changes your life irrevocably. And while it's truly wonderful having kids, there are a lot of things that are easier to do before you have them than after. Many of which will make you a better parent when you do have kids. And since you can delay pushing the button for a while, most people in rich countries do.

Yet when it comes to startups, a lot of people seem to think they're supposed to start them while they're still in college. Are you crazy? And what are the universities thinking? They go out of their way to ensure their students are well supplied with contraceptives, and yet they're setting up entrepreneurship programs and startup incubators left and right.

To be fair, the universities have their hand forced here. A lot

of incoming students are interested in startups. Universities are, at least de facto, expected to prepare them for their careers. So students who want to start startups hope universities can teach them about startups. And whether universities can do this or not, there's some pressure to claim they can, lest they lose applicants to other universities that do.

Can universities teach students about startups? Yes and no. They can teach students about startups, but as I explained before, this is not what you need to know. What you need to learn about are the needs of your own users, and you can't do that until you actually start the company.

[5]

So starting a startup is intrinsically something you can only really learn by doing it. And it's impossible to do that in college, for the reason I just explained: startups take over your life. You can't start a startup for real as a student, because if you start a startup for real you're not a student anymore. You may be nominally a student for a bit, but you won't even be that for long.

[6]

Given this dichotomy, which of the two paths should you take? Be a real student and not start a startup, or start a real startup and not be a student? I can answer that one for you. Do not start a startup in college. How to start a startup is just a subset of a bigger problem you're trying to solve: how to have a good life. And though starting a startup can be part of a good life for a lot of ambitious people, age 20 is not the optimal time to do it. Starting a startup is like a brutally fast depth-first search. Most people should still be searching breadth-first at 20.

You can do things in your early 20s that you can't do as well before or after, like plunge deeply into projects on a whim and travel super cheaply with no sense of a deadline. For unambitious people, this sort of thing is the dreaded "failure to launch," but for the ambitious ones it can be an incomparably valuable sort of exploration. If you start a startup at 20 and you're sufficiently successful, you'll never get to do it.

[7]

Mark Zuckerberg will never get to bum around a foreign country. He can do other things most people can't, like charter jets to fly him to foreign countries. But success has taken a lot of the serendipity out of his life. Facebook is running him as much as he's running Facebook. And while it can be very cool to be in the grip of a project you consider your life's work, there are advantages to serendipity too, especially early in life. Among other things it gives you more options to choose your life's work from.

There's not even a tradeoff here. You're not sacrificing anything if you forgo starting a startup at 20, because you're more likely to succeed if you wait. In the unlikely case that you're 20 and one of your side projects takes off like Facebook did, you'll face a choice of running with it or not, and it may be reasonable to run with it. But the usual way startups take off is for the founders to make them take off, and it's gratuitously stupid to do that at 20.

Try

Should you do it at any age? I realize I've made startups sound pretty hard. If I haven't, let me try again: starting a startup is really hard. What if it's too hard? How can you tell if you're up to this challenge?

The answer is the fifth counterintuitive point: you can't tell. Your life so far may have given you some idea what your prospects might be if you tried to become a mathematician, or a professional football player. But unless you've had a very strange life you haven't done much that was like being a startup founder.

Starting a startup will change you a lot. So what you're trying to estimate is not just what you are, but what you could grow into, and who can do that?

For the past 9 years it was my job to predict whether people would have what it took to start successful startups. It was easy to tell how smart they were, and most people reading this will be over that threshold. The hard part was predicting how tough and ambitious they would become. There may be no one who has more experience at trying to predict that,

so I can tell you how much an expert can know about it, and the answer is: not much. I learned to keep a completely open mind about which of the startups in each batch would turn out to be the stars.

The founders sometimes think they know. Some arrive feeling sure they will ace Y Combinator just as they've aced every one of the (few, artificial, easy) tests they've faced in life so far. Others arrive wondering how they got in, and hoping YC doesn't discover whatever mistake caused it to accept them. But there is little correlation between founders' initial attitudes and how well their companies do.

I've read that the same is true in the military — that the swaggering recruits are no more likely to turn out to be really tough than the quiet ones. And probably for the same reason: that the tests involved are so different from the ones in their previous lives.

If you're absolutely terrified of starting a startup, you probably shouldn't do it. But if you're merely unsure whether you're up to it, the only way to find out is to try. Just not now.

Ideas

So if you want to start a startup one day, what should you do in college? There are only two things you need initially: an idea and cofounders. And the m.o. for getting both is the same. Which leads to our sixth and last counterintuitive point: that the way to get startup ideas is not to try to think of startup ideas.

I've written a whole essay on this, so I won't repeat it all here. But the short version is that if you make a conscious effort to think of startup ideas, the ideas you come up with will not merely be bad, but bad and plausible-sounding, meaning you'll waste a lot of time on them before realizing they're bad.

The way to come up with good startup ideas is to take a step back. Instead of making a conscious effort to think of startup ideas, turn your mind into the type that startup ideas form in without any

conscious effort. In fact, so unconsciously that you don't even realize at first that they're startup ideas.

This is not only possible, it's how Apple, Yahoo, Google, and Facebook all got started. None of these companies were even meant to be companies at first. They were all just side projects. The best startups almost have to start as side projects, because great ideas tend to be such outliers that your conscious mind would reject them as ideas for companies.

Ok, so how do you turn your mind into the type that startup ideas form in unconsciously? (1) Learn a lot about things that matter, then (2) work on problems that interest you (3) with people you like and respect. The third part, incidentally, is how you get cofounders at the same time as the idea.

The first time I wrote that paragraph, instead of "learn a lot about things that matter," I wrote "become good at some technology." But that prescription, though sufficient, is too narrow. What was special about Brian Chesky and Joe Gebbia was not that they were experts in technology. They were good at design, and perhaps even more importantly, they were good at organizing groups and making projects happen. So you don't have to work on technology per se, so long as you work on problems demanding enough to stretch you.

What kind of problems are those? That is very hard to answer in the general case. History is full of examples of young people who were working on important problems that no one else at the time thought were important, and in particular that their parents didn't think were important. On the other hand, history is even fuller of examples of parents who thought their kids were wasting their time and who were right. So how do you know when you're working on real stuff?

[8]

I know how I know. Real problems are interesting, and I am self-indulgent in the sense that I always want to work on interesting things, even if no one else cares about them (in fact, especially if no one else cares about them), and find it very hard to make myself work on boring things, even if they're supposed to be

important.

My life is full of case after case where I worked on something just because it seemed interesting, and it turned out later to be useful in some worldly way. Y

Combinator itself was something I only did because it seemed interesting. So I seem to have some sort of internal compass that helps me out. But I don't know what other people have in their heads. Maybe if I think more about this I can come up with heuristics for recognizing genuinely interesting problems, but for the moment the best I can offer is the hopelessly question-begging advice that if you have a taste for genuinely interesting problems, indulging it energetically is the best way to prepare yourself for a startup. And indeed, probably also the best way to live.

[9]

But although I can't explain in the general case what counts as an interesting problem, I can tell you about a large subset of them. If you think of technology as something that's spreading like a sort of fractal stain, every moving point on the edge represents an interesting problem. So one guaranteed way to turn your mind into the type that has good startup ideas is to get yourself to the leading edge of some technology — to cause yourself, as Paul Buchheit put it, to "live in the future." When you reach that point, ideas that will seem to other people uncannily prescient will seem obvious to you. You may not realize they're startup ideas, but you'll know they're something that ought to exist.

For example, back at Harvard in the mid 90s a fellow grad student of my friends Robert and Trevor wrote his own voice over IP software. He didn't mean it to be a startup, and he never tried to turn it into one. He just wanted to talk to his girlfriend in Taiwan without paying for long distance calls, and since he was an expert on networks it seemed obvious to him that the way to do it was turn the sound into packets and ship it over the Internet. He never did any more with his software than talk to his girlfriend, but this is exactly the way the best startups get started.

So strangely enough the optimal thing to do in college if you want to be a successful startup founder is not some sort of new, vocational

version of college focused on "entrepreneurship." It's the classic version of college as education for its own sake. If you want to start a startup after college, what you should do in college is learn powerful things. And if you have genuine intellectual curiosity, that's what you'll naturally tend to do if you just follow your own inclinations.

[10]

The component of entrepreneurship that really matters is domain expertise. The way to become Larry Page was to become an expert on search. And the way to become an expert on search was to be driven by genuine curiosity, not some ulterior motive.

At its best, starting a startup is merely an ulterior motive for curiosity. And you'll do it best if you introduce the ulterior motive toward the end of the process.

So here is the ultimate advice for young would-be startup founders, boiled down to two words: just learn.

Notes

[1]

Some founders listen more than others, and this tends to be a predictor of success. One of the things I remember about the Airbnbs during YC is how intently they listened.

[2]

In fact, this is one of the reasons startups are possible. If big companies weren't plagued by internal inefficiencies, they'd be proportionately more effective, leaving less room for startups.

[3]

In a startup you have to spend a lot of time on schleps, but this sort of work is merely

unglamorous, not bogus.

[4]

What should you do if your true calling is gaming the system?

Management consulting.

[5]

The company may not be incorporated, but if you start to get significant numbers of users, you've started it, whether you realize it yet or not.

[6]

It shouldn't be that surprising that colleges can't teach students how to be good startup founders, because they can't teach them how to be good employees either.

The way universities "teach" students how to be employees is to hand off the task to companies via internship programs. But you couldn't do the equivalent thing for startups, because by definition if the students did well they would never come back.

[7]

Charles Darwin was 22 when he received an invitation to travel aboard the HMS Beagle as a naturalist. It was only because he was otherwise unoccupied, to a degree that alarmed his family, that he could accept it. And yet if he hadn't we probably would not know his name.

[8]

Parents can sometimes be especially conservative in this department. There are some whose definition of important problems includes only those on the critical path to med school.

[9]

I did manage to think of a heuristic for detecting whether you have a taste for interesting ideas: whether you find known boring ideas intolerable. Could you endure studying literary theory, or working in middle management at a large company?

[10]

In fact, if your goal is to start a startup, you can stick even more closely to the ideal of a liberal education than past generations have. Back when students focused mainly on getting a job after college, they thought at least a little about how the courses they took might look to an employer. And perhaps even worse, they might shy away from taking a difficult class lest they get a low grade, which would harm their all-important GPA. Good news: users don't care what your GPA was. And I've never heard of investors caring either. Y Combinator certainly never asks what classes you took in college or what grades you got in them.

Thanks to Sam Altman, Paul Buchheit, John Collison, Patrick Collison, Jessica Livingston, Robert Morris, Geoff Ralston, and Fred Wilson for reading drafts of this.

|