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

PAUL GRAHAM

What Startups Are Really Like

October 2009

(This essay is derived from a talk at the 2009 Startup School.)

I wasn't sure what to talk about at Startup School, so I decided to ask the founders of the startups we'd funded. What hadn't I written about yet?

I'm in the unusual position of being able to test the essays I write about startups. I hope the ones on other topics are right, but I have no way to test them. The ones on startups get tested by about 70 people every 6 months.

So I sent all the founders an email asking what surprised them about starting a startup. This amounts to asking what I got wrong, because if I'd explained things well enough, nothing should have surprised them.

I'm proud to report I got one response saying:

> What surprised me the most is that everything was actually fairly predictable!

The bad news is that I got over 100 other responses listing the surprises they encountered.

There were very clear patterns in the responses; it was remarkable how often several people had been surprised by exactly the same thing. These were the biggest:

1. Be Careful with Cofounders

This was the surprise mentioned by the most founders. There were two types of responses: that you have to be careful who you pick as a cofounder, and that you have to work hard to maintain your relationship.

What people wished they'd paid more attention to when choosing cofounders was character and commitment, not ability. This was particularly true with startups that failed. The lesson: don't pick cofounders who will flake.

Here's a typical reponse:

> You haven't seen someone's true colors unless you've worked with them on a startup.

The reason character is so important is that it's tested more severely than in most other situations. One founder said explicitly that the relationship between founders was more important than ability:

> I would rather cofound a startup with a friend than a stranger with higher output. Startups are so hard and emotional that the bonds and emotional and social support that come with friendship outweigh the extra output lost.

We learned this lesson a long time ago. If you look at the YC application, there are more questions about the commitment and relationship of the founders than their ability.

Founders of successful startups talked less about choosing cofounders and more about how hard they worked to maintain their relationship.

> One thing that surprised me is how the relationship of startup founders goes from a friendship to a marriage. My relationship with my cofounder went from just being friends to seeing each other all the time, fretting over the finances and cleaning up shit. And the startup was our baby. I summed it up once like this: "It's like we're married, but we're not fucking."

Several people used that word "married." It's a far more intense relationship than you usually see between coworkers—partly because the stresses are so much greater, and partly because at first the founders are the whole company. So this relationship has to be built of top quality materials and carefully maintained. It's the basis of everything.

2. Startups Take Over Your Life

Just as the relationship between cofounders is more intense than it usually is between coworkers, so is the relationship between the founders and the company. Running a startup is not like having a job or being a student, because it never stops. This is so foreign to most people's experience that they don't get it till it happens. [1]

> I didn't realize I would spend almost every waking moment either working or thinking about our startup. You enter a whole different way of life when it's your company vs. working for someone else's company.

It's exacerbated by the fast pace of startups, which makes it seem like time slows down:

> I think the thing that's been most surprising to me is how one's perspective on time shifts. Working on our startup, I remember time seeming to stretch out, so that a month was a huge interval.

In the best case, total immersion can be exciting:

> It's surprising how much you become consumed by your startup, in that you think about it day and night,

> but never once does it feel like "work."

Though I have to say, that quote is from someone we funded this summer. In a couple years he may not sound so chipper.

3. It's an Emotional Roller-coaster

This was another one lots of people were surprised about. The ups and downs were more extreme than they were prepared for.

In a startup, things seem great one moment and hopeless the next. And by next, I mean a couple hours later.

> The emotional ups and downs were the biggest surprise for me. One day, we'd think of ourselves as the next Google and dream of buying islands; the next, we'd be pondering how to let our loved ones know of our utter failure; and on and on.

The hard part, obviously, is the lows. For a lot of founders that was the big surprise:

> How hard it is to keep everyone motivated during rough days or weeks, i.e. how low the lows can be.

After a while, if you don't have significant success to cheer you up, it wears you out:

> Your most basic advice to founders is "just don't die," but the energy to keep a company going in lieu of unburdening success isn't free; it is siphoned from the founders themselves.

There's a limit to how much you can take. If you get to the point where you can't keep working anymore, it's not the end of the world. Plenty of famous founders have had some failures along the way.

4. It Can Be Fun

The good news is, the highs are also very high. Several founders said what surprised them most about doing a startup was how fun it was:

> I think you've left out just how fun it is to do a startup. I am more fulfilled in my work than pretty much any of my friends who did not start companies.

What they like most is the freedom:

> I'm surprised by how much better it feels to be working on something that is challenging and creative, something I believe in, as opposed to the hired-gun stuff I was doing before. I knew it would feel better; what's surprising is how much better.

Frankly, though, if I've misled people here, I'm not eager to fix that. I'd rather have everyone think starting a startup is grim and hard than have founders go into it expecting it to be fun, and a few months later saying "This is supposed to be *fun*? Are you kidding?"

The truth is, it wouldn't be fun for most people. A lot of what we try to do in the application process is to weed out the people who wouldn't like it, both for our sake and theirs.

The best way to put it might be that starting a startup is fun the way a survivalist training course would be fun, if you're into that sort of thing. Which is to say, not at all, if you're not.

5. Persistence Is the Key

A lot of founders were surprised how important persistence was in startups. It was both a negative and a positive surprise: they were surprised both by the degree of persistence required

> Everyone said how determined and resilient you must be, but going through it made me realize that the determination required was still understated.

and also by the degree to which persistence alone was able to dissolve obstacles:

> If you are persistent, even problems that seem out of your control (i.e. immigration) seem to work themselves out.

Several founders mentioned specifically how much more important persistence was than intelligence.

> I've been surprised again and again by just how much more important persistence is than raw intelligence.

This applies not just to intelligence but to ability in general, and that's why so many people said character was more important in choosing cofounders.

6. Think Long-Term

You need persistence because everything takes longer than you expect. A lot of people were surprised by that.

> I'm continually surprised by how long everything can take. Assuming your product doesn't experience the explosive growth that very few products do, everything from development to dealmaking (especially dealmaking) seems to take 2-3x longer than I always imagine.

One reason founders are surprised is that because they work fast, they expect everyone else to. There's a shocking amount of shear stress at every point where a startup touches a more bureaucratic organization, like a big company or a VC fund. That's why fundraising and the enterprise market kill and maim so many startups. [2]

But I think the reason most founders are surprised by how long it takes is that they're overconfident. They think they're going to be an instant success, like YouTube or Facebook. You tell them only 1 out of 100 successful startups has a trajectory like that, and they all think "we're going to be that 1."

Maybe they'll listen to one of the more successful founders:

> The top thing I didn't understand before going into it is that persistence is the name of the game. For the vast majority of startups that become successful, it's going to be a *really* long journey, at least 3 years and probably 5+.

There is a positive side to thinking longer-term. It's not just that you have to resign yourself to everything taking longer than it should. If you work patiently it's less stressful, and you can do better work:

> Because we're relaxed, it's so much easier to have fun doing what we do. Gone is the awkward nervous energy fueled by the desperate need to not fail guiding our actions. We can concentrate on doing what's best for our company, product, employees and customers.

That's why things get so much better when you hit ramen profitability. You can shift into a different mode of working.

7. Lots of Little Things

We often emphasize how rarely startups win simply because they hit on some magic idea. I think founders have now gotten that into their heads. But a lot were surprised to find this also applies within startups. You have to do lots of different things:

> It's much more of a grind than glamorous. A timeslice selected at random would more likely find me tracking down a weird DLL loading bug on Swedish Windows, or tracking down a bug in the financial model Excel spreadsheet the night before a board meeting, rather than having brilliant flashes of strategic insight.

Most hacker-founders would like to spend all their time programming. You won't get to, unless you fail. Which can be transformed into: If you spend all your time programming, you will fail.

The principle extends even into programming. There is rarely a single brilliant hack that ensures success:

> I learnt never to bet on any one feature or deal or anything to bring you success. It is never a single thing. Everything is just incremental and you just have to keep doing lots of those things until you strike something.

Even in the rare cases where a clever hack makes your fortune, you probably won't know till later:

> There is no such thing as a killer feature. Or at least you won't know what it is.

So the best strategy is to try lots of different things. The reason not to put all your eggs in one basket is not the usual one, which applies even when you know which basket is best. In a startup you don't even know that.

8. Start with Something Minimal

Lots of founders mentioned how important it was to launch with the simplest possible thing. By this point everyone knows you should release fast and iterate. It's practically a mantra at YC. But even so a lot of people seem to have been burned by not doing it:

> Build the absolute smallest thing that can be considered a complete application and ship it.

Why do people take too long on the first version? Pride, mostly. They hate to release something that could be better. They worry what people will say about them. But you have to overcome this:

> Doing something "simple" at first glance does not mean you aren't doing something meaningful, defensible, or valuable.

Don't worry what people will say. If your first version is so impressive that trolls don't make fun of it, you waited too long to launch. [3]

One founder said this should be your approach to all programming, not just startups, and I tend to agree.

> Now, when coding, I try to think "How can I write this such that if people saw my code, they'd be amazed at how little there is and how little it does?"

Over-engineering is poison. It's not like doing extra work for extra credit. It's more like telling a lie that you then have to remember so you don't contradict it.

9. Engage Users

Product development is a conversation with the user that doesn't really start till you launch. Before you launch, you're like a police artist before he's shown the first version of his sketch to the witness.

It's so important to launch fast that it may be better to think of your initial version not as a product, but as a trick for getting users to start talking to you.

> I learned to think about the initial stages of a startup as a giant experiment. All products should be considered experiments, and those that have a market show promising results extremely quickly.

Once you start talking to users, I guarantee you'll be surprised by what they tell you.

> When you let customers tell you what they're after, they will often reveal amazing details about what they find valuable as well what they're willing to pay for.

The surprise is generally positive as well as negative. They won't like what you've built, but there will be other things they would like that would be trivially easy to implement. It's not till you start the conversation by launching the wrong thing that they can express (or perhaps even realize) what they're looking for.

10. Change Your Idea

To benefit from engaging with users you have to be willing to change your idea. We've always encouraged founders to see a startup idea as a hypothesis rather than a blueprint. And yet they're still surprised how well it works to change the idea.

> Normally if you complain about something being hard, the general advice is to work harder. With a startup, I think you should find a problem that's easy for you to solve. Optimizing in solution-space is familiar and straightforward, but you can make enormous gains playing around in problem-space.

Whereas mere determination, without flexibility, is a greedy algorithm that may get you nothing more than a mediocre local maximum:

> When someone is determined, there's still a danger that they'll follow a long, hard path that ultimately leads nowhere.

You want to push forward, but at the same time twist and turn to find the most promising path. One founder put it very succinctly:

> Fast iteration is the key to success.

One reason this advice is so hard to follow is that people don't realize how hard it is to judge startup ideas, particularly their own. Experienced founders learn to keep an open mind:

> Now I don't laugh at ideas anymore, because I realized how terrible I was at knowing if they were good or not.

You can never tell what will work. You just have to do whatever seems best at each point. We do this with YC itself. We still don't know if it will work, but it seems like a decent hypothesis.

11. Don't Worry about Competitors

When you think you've got a great idea, it's sort of like having a guilty conscience about something. All someone has to do is look at you funny, and you think "Oh my God, *they know.*"

These alarms are almost always false:

> Companies that seemed like competitors and threats at first glance usually never were when you really looked at it. Even if they were operating in the same area, they had a different goal.

One reason people overreact to competitors is that they overvalue ideas. If ideas really were the key, a competitor with the same idea would be a real threat. But it's usually execution that matters:

> All the scares induced by seeing a new competitor pop up are forgotten weeks later. It always comes down to your own product and approach to the market.

This is generally true even if competitors get lots of attention.

> Competitors riding on lots of good blogger perception aren't really the winners and can disappear from the map quickly. You need consumers after all.

Hype doesn't make satisfied users, at least not for something as complicated as technology.

12. It's Hard to Get Users

A lot of founders complained about how hard it was to get users, though.

> I had no idea how much time and effort needed to go into attaining users.

This is a complicated topic. When you can't get users, it's hard to say whether the problem is lack of exposure, or whether the product's simply bad. Even good products can be blocked by switching or integration costs:

> Getting people to use a new service is incredibly difficult. This is especially true for a service that other companies can use, because it requires their developers to do work. If you're small, they don't think it is urgent. [4]

The sharpest criticism of YC came from a founder who said we didn't focus enough on customer acquisition:

> YC preaches "make something people want" as an engineering task, a never ending stream of feature after feature until enough people are happy and the application takes off. There's very little focus on the cost of customer acquisition.

This may be true; this may be something we need to fix, especially for applications like games. If you make something where the challenges are mostly technical, you can rely on word of mouth, like Google did. One founder was surprised by how well that worked for him:

> There is an irrational fear that no one will buy your product. But if you work hard and incrementally make it better, there is no need to worry.

But with other types of startups you may win less by features and more by deals and marketing.

13. Expect the Worst with Deals

Deals fall through. That's a constant of the startup world. Startups are powerless, and good startup ideas generally seem wrong. So everyone is nervous about closing deals with you, and you have no way to make them.

This is particularly true with investors:

> In retrospect, it would have been much better if we had operated under the assumption that we would never get any additional outside investment. That would have focused us on finding revenue streams early.

My advice is generally pessimistic. Assume you won't get money, and if someone does offer you any, assume you'll never get any more.

> If someone offers you money, take it. You say it a lot, but I think it needs even more emphasizing. We had the opportunity to raise a lot more money than we did last year and I wish we had.

Why do founders ignore me? Mostly because they're optimistic by nature. The mistake is to be optimistic about things you can't control. By all means be optimistic about your ability to make something great. But you're asking for trouble if you're optimistic about big companies or investors.

14. Investors Are Clueless

A lot of founders mentioned how surprised they were by the cluelessness of investors:

> They don't even know about the stuff they've invested in. I met some investors that had invested in a hardware device and when I asked them to demo the device they had difficulty switching it on.

Angels are a bit better than VCs, because they usually have startup experience themselves:

> VC investors don't know half the time what they are talking about and are years behind in their thinking. A few were great, but 95% of the investors we dealt with were unprofessional, didn't seem to be very good at business or have any kind of creative vision. Angels were generally much better to talk to.

Why are founders surprised that VCs are clueless? I think it's because they seem so formidable.

The reason VCs seem formidable is that it's their profession to. You get to be a VC by convincing asset managers to trust you with hundreds of millions of dollars. How do you do that? You have to seem confident, and you have to seem like you understand technology. [5]

15. You May Have to Play Games

Because investors are so bad at judging you, you have to work harder than you should at selling yourself. One founder said the thing that surprised him most was

> The degree to which feigning certitude impressed investors.

This is the thing that has surprised *me* most about YC founders' experiences. This summer we invited some of the alumni to talk to the new startups about fundraising, and pretty much 100% of their advice was about investor psychology. I thought I was cynical about VCs, but the founders were much more cynical.

> A lot of what startup founders do is just posturing. It works.

VCs themselves have no idea of the extent to which the startups they like are the ones that are best at selling themselves to VCs. [6] It's exactly the same phenomenon we saw a step earlier. VCs get money by seeming confident to LPs, and founders get money by seeming confident to VCs.

16. Luck Is a Big Factor

With two such random linkages in the path between startups and money, it shouldn't be surprising that luck is a big factor in deals. And yet a lot of founders are surprised by it.

> I didn't realize how much of a role luck plays and how much is outside of our control.

If you think about famous startups, it's pretty clear how big a role luck plays. Where would Microsoft be if IBM insisted on an exclusive license for DOS?

Why are founders fooled by this? Business guys probably aren't, but hackers are used to a world where skill is paramount, and you get what you deserve.

> When we started our startup, I had bought the hype of the startup founder dream: that this is a game of skill. It is, in some ways. Having skill is valuable. So is being determined as all hell. But being lucky is the critical ingredient.

Actually the best model would be to say that the outcome is the *product* of skill, determination, and luck. No matter how much skill and determination you have, if you roll a zero for luck, the outcome is zero.

These quotes about luck are not from founders whose startups failed. Founders who fail quickly tend to blame themselves. Founders who succeed quickly don't usually realize how lucky they were. It's the ones in the middle who see how important luck is.

17. The Value of Community

A surprising number of founders said what surprised them most about starting a startup was the value of community. Some meant the micro-community of YC founders:

> The immense value of the peer group of YC companies, and facing similar obstacles at similar times.

which shouldn't be that surprising, because that's why it's structured that way. Others were surprised at the value of the startup community in the larger sense:

> How advantageous it is to live in Silicon Valley, where you can't help but hear all the cutting-edge tech and startup news, and run into useful people constantly.

The specific thing that surprised them most was the general spirit of benevolence:

> One of the most surprising things I saw was the willingness of people to help us. Even people who had nothing to gain went out of their way to help our startup succeed.

and particularly how it extended all the way to the top:

> The surprise for me was how accessible important and interesting people are. It's amazing how easily you can reach out to people and get immediate feedback.

This is one of the reasons I like being part of this world. Creating wealth is not a zero-sum game, so you don't have to stab people in the back to win.

18. You Get No Respect

There was one surprise founders mentioned that I'd forgotten about: that outside the startup world, startup founders get no respect.

> In social settings, I found that I got a lot more respect when I said, "I worked on Microsoft Office" instead of "I work at a small startup you've never heard of called x."

Partly this is because the rest of the world just doesn't get startups, and partly it's yet another consequence of the fact that most good startup ideas seem bad:

> If you pitch your idea to a random person, 95% of the time you'll find the person instinctively thinks the idea will be a flop and you're wasting your time (although they probably won't say this directly).

Unfortunately this extends even to dating:

> It surprised me that being a startup founder does not get you more admiration from women.

I did know about that, but I'd forgotten.

19. Things Change as You Grow

The last big surprise founders mentioned is how much things changed as they grew. The biggest change was that you got to program even less:

> Your job description as technical founder/CEO is completely rewritten every 6-12 months. Less coding, more managing/planning/company building, hiring, cleaning up messes, and generally getting things in place for what needs to happen a few months from now.

In particular, you now have to deal with employees, who often have different motivations:

> I knew the founder equation and had been focused on it since I knew I wanted to start a startup as a 19 year old. The employee equation is quite different so it took me a while to get it down.

Fortunately, it can become a lot less stressful once you reach cruising altitude:

> I'd say 75% of the stress is gone now from when we first started. Running a business is so much more enjoyable now. We're more confident. We're more patient. We fight less. We sleep more.

I wish I could say it was this way for every startup that succeeded, but 75% is probably on the high side.

The Super-Pattern

There were a few other patterns, but these were the biggest. One's first thought when looking at them all is to ask if there's a super-pattern, a pattern to the patterns.

I saw it immediately, and so did a YC founder I read the list to. These are supposed to be the surprises, the things I didn't tell people. What do they all have in common? They're all things I tell people. If I wrote a new essay with the same outline as this that wasn't summarizing the founders' responses, everyone would say I'd run out of ideas and was just repeating myself.

What is going on here?

When I look at the responses, the common theme is that starting a startup was like I said, but way more so. People just don't seem to get how different it is till they do it. Why? The key to that mystery is to ask, how different *from what?* Once you phrase it that way, the answer is obvious: from a job. Everyone's model of work is a job. It's completely pervasive. Even if you've never had a job, your parents probably did, along with practically every other adult you've met.

Unconsciously, everyone expects a startup to be like a job, and that explains most of the surprises. It explains why people are surprised how carefully you have to choose cofounders and how hard you have to work to maintain your relationship. You don't have to do that with coworkers. It explains why the ups and downs are surprisingly extreme. In a job there is much more damping. But it also explains why the good times are surprisingly good: most people can't imagine such freedom. As you go down the list, almost all the surprises are surprising in how much a startup differs from a job.

You probably can't overcome anything so pervasive as the model of work you grew up with. So the best solution is to be consciously aware of that. As you go into a startup, you'll be thinking "everyone says it's really extreme." Your next thought will probably be "but I can't believe it will be that bad." If you want to avoid being surprised, the next thought after that should be: "and the reason I can't believe it will be that bad is that my model of work is a job."

Notes

[1] Graduate students might understand it. In grad school you always feel you should be working on your thesis. It doesn't end every semester like classes do.

[2] The best way for a startup to engage with slow-moving organizations is to fork off separate processes to deal with them. It's when they're on the critical path that they kill you—when you depend on closing a deal to move forward. It's worth taking extreme measures to avoid that.

[3] This is a variant of Reid Hoffman's principle that if you aren't embarrassed by what you launch with, you waited too long to launch.

[4] The question to ask about what you've built is not whether it's good, but whether it's good enough to supply the activation energy required.

[5] Some VCs seem to understand technology because they actually do, but that's overkill; the defining test is whether you can talk about it well enough to convince limited partners.

[6] This is the same phenomenon you see with defense contractors or fashion brands. The dumber the customers, the more effort you expend on the process of selling things to them rather than making the things you sell.

Thanks: to Jessica Livingston for reading drafts of this, and to all the founders who responded to my email.

Related:

- Startups in 13 Sentences
- The Hardest Lessons for Startups to Learn
- How Not to Die
- The 18 Mistakes That Kill Startups
- A Fundraising Survival Guide
- Russian Translation
- Korean Translation
- Hebrew Translation