

A collection of Paul Graham writingsEssays

Paul Graham

2024-04-23T07:10:50Z

Contents

064 The Power of the Marginal	3
069 The 18 Mistakes That Kill Startups	15
089 You Weren't Meant to Have a Boss	27
096 Cities and Ambition	32
097 The Pooled-Risk Company Management Company	39
100 The Other Half of "Artists Ship"	42
105 Startups in 13 Sentences	45
110 Relentlessly Resourceful	48
115 Maker's Schedule, Manager's Schedule	50
116 Ramen Profitable	53
119 The Anatomy of Determination	57
125 Organic Startup Ideas	60
126 How to Lose Time and Money	62
127 The Top Idea in Your Mind	63
128 The Acceleration of Addictiveness	66
134 What We Look for in Founders	69
141 Schlep Blindness	70

146 Writing and Speaking	72
147 The Top of My Todo List	74
148 Black Swan Farming	75
164 What Doesn't Seem Like Work?	79
169 Default Alive or Default Dead?	80
170 Write Like You Talk	83
175 Life is Short	85
182 The Lesson to Unlearn	88
196 Billionaires Build	96
197 Earnestness	103
200 Write Simply	106
207 How to Work Hard	107
210 Is There Such a Thing as Good Taste?	114
216 The Need to Read	116
217 How to Get New Ideas	117
218 How to Do Great Work	117
219 Superlinear Returns	141



1. Title Page
2. Cover
3. Table of Contents

064 The Power of the Marginal

Want to start a startup? Get funded by Y Combinator.

June 2006

(This essay is derived from talks at Usenix 2006 and Railsconf 2006.)

A couple years ago my friend Trevor and I went to look at the Apple garage. As we stood there, he said that as a kid growing up in Saskatchewan he'd been amazed at the dedication Jobs and Wozniak must have had to work in a garage.

"Those guys must have been freezing!"

That's one of California's hidden advantages: the mild climate means there's lots of marginal space. In cold places that margin gets trimmed off. There's a sharper line between outside and inside, and only projects that are officially sanctioned -- by organizations, or parents, or wives, or at least by oneself -- get proper indoor space. That raises the activation energy for new ideas. You can't just tinker. You have to justify.

Some of Silicon Valley's most famous companies began in garages: Hewlett-Packard in 1938, Apple in 1976, Google in 1998. In Apple's case the garage story is a bit of an urban legend. Woz says all they did there was assemble some computers, and that he did all the actual design of the Apple I and Apple II in his apartment or his cube at HP. ¹ This was apparently too marginal even for Apple's PR people.

By conventional standards, Jobs and Wozniak were marginal people too. Obviously they were smart, but they can't have looked good on paper. They were at the time a pair of college dropouts with about three years of school between them, and hippies to boot. Their previous business experience consisted of making "blue boxes" to hack into the phone system, a business with the rare distinction of being both illegal and unprofitable.

Outsiders

Now a startup operating out of a garage in Silicon Valley would feel part of an exalted tradition, like the poet in his garret, or the painter who can't afford to heat his studio and thus has to wear a beret indoors. But in 1976 it didn't seem so cool. The world hadn't yet realized that starting a computer company was in the same category as being a writer or a painter. It hadn't been for long. Only in the preceding couple years had the dramatic fall in the cost of hardware allowed outsiders to compete.

In 1976, everyone looked down on a company operating out of a garage, including the founders. One of the first things Jobs did when they got some money was to rent office space. He wanted Apple to seem like a real company.

They already had something few real companies ever have: a fabulously well designed product. You'd think they'd have had more confidence. But I've talked to a lot of startup founders, and it's always this way. They've built something that's going to change the world, and they're worried about some nit like not having proper business cards.

That's the paradox I want to explore: great new things often come from the margins, and yet the people who discover them are looked down on by everyone, including themselves.

It's an old idea that new things come from the margins. I want to examine its internal structure. Why do great ideas come from the margins? What kind of ideas? And is there anything we can do to encourage the process?

Insiders

One reason so many good ideas come from the margin is simply that there's so much of it. There have to be more outsiders than insiders, if insider means anything. If the number of outsiders is huge it will always seem as if a lot of ideas come from them, even if few do per capita. But I think there's more going on than this. There are real disadvantages to being an insider, and in some kinds of work they can outweigh the advantages.

Imagine, for example, what would happen if the government decided to commission someone to write an official Great American Novel. First there'd be a huge ideological squabble over who to choose. Most of the best writers would be excluded for having offended one side or the other. Of the remainder, the smart ones would refuse such a job, leaving only a few with the wrong sort of ambition. The committee would choose one at the height of his career -- that is, someone whose best work was behind him -- and hand over the project with copious free advice about how the book should show in positive terms the strength and diversity of the American people, etc, etc.

The unfortunate writer would then sit down to work with a huge weight of expectation on his shoulders. Not wanting to blow such a public commission, he'd play it safe. This book had better command respect, and the way to ensure that would be to make it a tragedy. Audiences have to be enticed to laugh, but if you kill people they feel obliged to take you seriously. As everyone knows, America plus tragedy equals the Civil War, so that's what it would have to be about. When finally completed twelve years later, the book would be a 900-page pastiche of existing popular novels -- roughly *Gone with the Wind* plus *Roots*. But its bulk and celebrity would make it a bestseller for a few months, until blown out of the water by a talk-show host's autobiography. The book would be made into a movie and thereupon forgotten, except by the more waspish sort of reviewers, among whom it would be a byword for bogusness like Milli Vanilli or *Battlefield Earth*.

Maybe I got a little carried away with this example. And yet is this not at each point the way such a project would play out? The government knows better than to get into the novel business, but in other fields where they have a natural monopoly, like nuclear waste dumps, aircraft carriers, and regime change, you'd find plenty of projects isomorphic to this one -- and indeed, plenty that were less successful.

This little thought experiment suggests a few of the disadvantages of insider projects: the selection of the wrong kind of people, the excessive scope, the inability to take risks, the need to seem serious, the weight of expectations, the power of vested interests, the undiscerning audience, and perhaps most dangerous, the tendency of such work to become a duty rather than a pleasure.

Tests

A world with outsiders and insiders implies some kind of test for distinguishing between them. And the trouble with most tests for selecting elites is that there

are two ways to pass them: to be good at what they try to measure, and to be good at hacking the test itself.

So the first question to ask about a field is how honest its tests are, because this tells you what it means to be an outsider. This tells you how much to trust your instincts when you disagree with authorities, whether it's worth going through the usual channels to become one yourself, and perhaps whether you want to work in this field at all.

Tests are least hackable when there are consistent standards for quality, and the people running the test really care about its integrity. Admissions to PhD programs in the hard sciences are fairly honest, for example. The professors will get whoever they admit as their own grad students, so they try hard to choose well, and they have a fair amount of data to go on. Whereas undergraduate admissions seem to be much more hackable.

One way to tell whether a field has consistent standards is the overlap between the leading practitioners and the people who teach the subject in universities. At one end of the scale you have fields like math and physics, where nearly all the teachers are among the best practitioners. In the middle are medicine, law, history, architecture, and computer science, where many are. At the bottom are business, literature, and the visual arts, where there's almost no overlap between the teachers and the leading practitioners. It's this end that gives rise to phrases like "those who can't do, teach."

Incidentally, this scale might be helpful in deciding what to study in college. When I was in college the rule seemed to be that you should study whatever you were most interested in. But in retrospect you're probably better off studying something moderately interesting with someone who's good at it than something very interesting with someone who isn't. You often hear people say that you shouldn't major in business in college, but this is actually an instance of a more general rule: don't learn things from teachers who are bad at them.

How much you should worry about being an outsider depends on the quality of the insiders. If you're an amateur mathematician and think you've solved a famous open problem, better go back and check. When I was in grad school, a friend in the math department had the job of replying to people who sent in proofs of Fermat's last theorem and so on, and it did not seem as if he saw it as a valuable source of tips -- more like manning a mental health hotline. Whereas if the stuff you're writing seems different from what English professors are interested in, that's not necessarily a problem.

Anti-Tests

Where the method of selecting the elite is thoroughly corrupt, most of the good people will be outsiders. In art, for example, the image of the poor, misunderstood genius is not just one possible image of a great artist: it's the *standard* image. I'm not saying it's correct, incidentally, but it is telling how well this image has stuck. You couldn't make a rap like that stick to math or

medicine. 2

If it's corrupt enough, a test becomes an anti-test, filtering out the people it should select by making them to do things only the wrong people would do. Popularity in high school seems to be such a test. There are plenty of similar ones in the grownup world. For example, rising up through the hierarchy of the average big company demands an attention to politics few thoughtful people could spare. 3 Someone like Bill Gates can grow a company under him, but it's hard to imagine him having the patience to climb the corporate ladder at General Electric -- or Microsoft, actually.

It's kind of strange when you think about it, because lord-of-the-flies schools and bureaucratic companies are both the default. There are probably a lot of people who go from one to the other and never realize the whole world doesn't work this way.

I think that's one reason big companies are so often blindsided by startups. People at big companies don't realize the extent to which they live in an environment that is one large, ongoing test for the wrong qualities.

If you're an outsider, your best chances for beating insiders are obviously in fields where corrupt tests select a lame elite. But there's a catch: if the tests are corrupt, your victory won't be recognized, at least in your lifetime. You may feel you don't need that, but history suggests it's dangerous to work in fields with corrupt tests. You may beat the insiders, and yet not do as good work, on an absolute scale, as you would in a field that was more honest.

Standards in art, for example, were almost as corrupt in the first half of the eighteenth century as they are today. This was the era of those fluffy idealized portraits of countesses with their lapdogs. Chardin decided to skip all that and paint ordinary things as he saw them. He's now considered the best of that period -- and yet not the equal of Leonardo or Bellini or Memling, who all had the additional encouragement of honest standards.

It can be worth participating in a corrupt contest, however, if it's followed by another that isn't corrupt. For example, it would be worth competing with a company that can spend more than you on marketing, as long as you can survive to the next round, when customers compare your actual products. Similarly, you shouldn't be discouraged by the comparatively corrupt test of college admissions, because it's followed immediately by less hackable tests. 4

Risk

Even in a field with honest tests, there are still advantages to being an outsider. The most obvious is that outsiders have nothing to lose. They can do risky things, and if they fail, so what? Few will even notice.

The eminent, on the other hand, are weighed down by their eminence. Eminence is like a suit: it impresses the wrong people, and it constrains the wearer.

Outsiders should realize the advantage they have here. Being able to take risks is hugely valuable. Everyone values safety too much, both the obscure and the eminent. No one wants to look like a fool. But it's very useful to be able to. If most of your ideas aren't stupid, you're probably being too conservative. You're not bracketing the problem.

Lord Acton said we should judge talent at its best and character at its worst. For example, if you write one great book and ten bad ones, you still count as a great writer -- or at least, a better writer than someone who wrote eleven that were merely good. Whereas if you're a quiet, law-abiding citizen most of the time but occasionally cut someone up and bury them in your backyard, you're a bad guy.

Almost everyone makes the mistake of treating ideas as if they were indications of character rather than talent -- as if having a stupid idea made you stupid. There's a huge weight of tradition advising us to play it safe. "Even a fool is thought wise if he keeps silent," says the Old Testament (Proverbs 17:28).

Well, that may be fine advice for a bunch of goatherds in Bronze Age Palestine. There conservatism would be the order of the day. But times have changed. It might still be reasonable to stick with the Old Testament in political questions, but materially the world now has a lot more state. Tradition is less of a guide, not just because things change faster, but because the space of possibilities is so large. The more complicated the world gets, the more valuable it is to be willing to look like a fool.

Delegation

And yet the more successful people become, the more heat they get if they screw up -- or even seem to screw up. In this respect, as in many others, the eminent are prisoners of their own success. So the best way to understand the advantages of being an outsider may be to look at the disadvantages of being an insider.

If you ask eminent people what's wrong with their lives, the first thing they'll complain about is the lack of time. A friend of mine at Google is fairly high up in the company and went to work for them long before they went public. In other words, he's now rich enough not to have to work. I asked him if he could still endure the annoyances of having a job, now that he didn't have to. And he said that there weren't really any annoyances, except -- and he got a wistful look when he said this -- that he got *so much email*.

The eminent feel like everyone wants to take a bite out of them. The problem is so widespread that people pretending to be eminent do it by pretending to be overstretched.

The lives of the eminent become scheduled, and that's not good for thinking. One of the great advantages of being an outsider is long, uninterrupted blocks of time. That's what I remember about grad school: apparently endless supplies of time, which I spent worrying about, but not writing, my dissertation. Obscurity is like health food -- unpleasant, perhaps, but good for you. Whereas fame tends

to be like the alcohol produced by fermentation. When it reaches a certain concentration, it kills off the yeast that produced it.

The eminent generally respond to the shortage of time by turning into managers. They don't have time to work. They're surrounded by junior people they're supposed to help or supervise. The obvious solution is to have the junior people do the work. Some good stuff happens this way, but there are problems it doesn't work so well for: the kind where it helps to have everything in one head.

For example, it recently emerged that the famous glass artist Dale Chihuly hasn't actually blown glass for 27 years. He has assistants do the work for him. But one of the most valuable sources of ideas in the visual arts is the resistance of the medium. That's why oil paintings look so different from watercolors. In principle you could make any mark in any medium; in practice the medium steers you. And if you're no longer doing the work yourself, you stop learning from this.

So if you want to beat those eminent enough to delegate, one way to do it is to take advantage of direct contact with the medium. In the arts it's obvious how: blow your own glass, edit your own films, stage your own plays. And in the process pay close attention to accidents and to new ideas you have on the fly. This technique can be generalized to any sort of work: if you're an outsider, don't be ruled by plans. Planning is often just a weakness forced on those who delegate.

Is there a general rule for finding problems best solved in one head? Well, you can manufacture them by taking any project usually done by multiple people and trying to do it all yourself. Wozniak's work was a classic example: he did everything himself, hardware and software, and the result was miraculous. He claims not one bug was ever found in the Apple II, in either hardware or software.

Another way to find good problems to solve in one head is to focus on the grooves in the chocolate bar -- the places where tasks are divided when they're split between several people. If you want to beat delegation, focus on a vertical slice: for example, be both writer and editor, or both design buildings and construct them.

One especially good groove to span is the one between tools and things made with them. For example, programming languages and applications are usually written by different people, and this is responsible for a lot of the worst flaws in programming languages. I think every language should be designed simultaneously with a large application written in it, the way C was with Unix.

Techniques for competing with delegation translate well into business, because delegation is endemic there. Instead of avoiding it as a drawback of senility, many companies embrace it as a sign of maturity. In big companies software is often designed, implemented, and sold by three separate types of people. In startups one person may have to do all three. And though this feels stressful, it's one reason startups win. The needs of customers and the means of satisfying

them are all in one head.

Focus

The very skill of insiders can be a weakness. Once someone is good at something, they tend to spend all their time doing that. This kind of focus is very valuable, actually. Much of the skill of experts is the ability to ignore false trails. But focus has drawbacks: you don't learn from other fields, and when a new approach arrives, you may be the last to notice.

For outsiders this translates into two ways to win. One is to work on a variety of things. Since you can't derive as much benefit (yet) from a narrow focus, you may as well cast a wider net and derive what benefit you can from similarities between fields. Just as you can compete with delegation by working on larger vertical slices, you can compete with specialization by working on larger horizontal slices -- by both writing and illustrating your book, for example.

The second way to compete with focus is to see what focus overlooks. In particular, new things. So if you're not good at anything yet, consider working on something so new that no one else is either. It won't have any prestige yet, if no one is good at it, but you'll have it all to yourself.

The potential of a new medium is usually underestimated, precisely because no one has yet explored its possibilities. Before Durer tried making engravings, no one took them very seriously. Engraving was for making little devotional images -- basically fifteenth century baseball cards of saints. Trying to make masterpieces in this medium must have seemed to Durer's contemporaries the way that, say, making masterpieces in comics might seem to the average person today.

In the computer world we get not new mediums but new platforms: the mini-computer, the microprocessor, the web-based application. At first they're always dismissed as being unsuitable for real work. And yet someone always decides to try anyway, and it turns out you can do more than anyone expected. So in the future when you hear people say of a new platform: yeah, it's popular and cheap, but not ready yet for real work, jump on it.

As well as being more comfortable working on established lines, insiders generally have a vested interest in perpetuating them. The professor who made his reputation by discovering some new idea is not likely to be the one to discover its replacement. This is particularly true with companies, who have not only skill and pride anchoring them to the status quo, but money as well. The Achilles heel of successful companies is their inability to cannibalize themselves. Many innovations consist of replacing something with a cheaper alternative, and companies just don't want to see a path whose immediate effect is to cut an existing source of revenue.

So if you're an outsider you should actively seek out contrarian projects. Instead of working on things the eminent have made prestigious, work on things that could steal that prestige.

The really juicy new approaches are not the ones insiders reject as impossible, but those they ignore as undignified. For example, after Wozniak designed the Apple II he offered it first to his employer, HP. They passed. One of the reasons was that, to save money, he'd designed the Apple II to use a TV as a monitor, and HP felt they couldn't produce anything so declassé.

Less

Wozniak used a TV as a monitor for the simple reason that he couldn't afford a monitor. Outsiders are not merely free but compelled to make things that are cheap and lightweight. And both are good bets for growth: cheap things spread faster, and lightweight things evolve faster.

The eminent, on the other hand, are almost forced to work on a large scale. Instead of garden sheds they must design huge art museums. One reason they work on big things is that they can: like our hypothetical novelist, they're flattered by such opportunities. They also know that big projects will by their sheer bulk impress the audience. A garden shed, however lovely, would be easy to ignore; a few might even snicker at it. You can't snicker at a giant museum, no matter how much you dislike it. And finally, there are all those people the eminent have working for them; they have to choose projects that can keep them all busy.

Outsiders are free of all this. They can work on small things, and there's something very pleasing about small things. Small things can be perfect; big ones always have something wrong with them. But there's a magic in small things that goes beyond such rational explanations. All kids know it. Small things have more personality.

Plus making them is more fun. You can do what you want; you don't have to satisfy committees. And perhaps most important, small things can be done fast. The prospect of seeing the finished project hangs in the air like the smell of dinner cooking. If you work fast, maybe you could have it done tonight.

Working on small things is also a good way to learn. The most important kinds of learning happen one project at a time. ("Next time, I won't...") The faster you cycle through projects, the faster you'll evolve.

Plain materials have a charm like small scale. And in addition there's the challenge of making do with less. Every designer's ears perk up at the mention of that game, because it's a game you can't lose. Like the JV playing the varsity, if you even tie, you win. So paradoxically there are cases where fewer resources yield better results, because the designers' pleasure at their own ingenuity more than compensates. 5

So if you're an outsider, take advantage of your ability to make small and inexpensive things. Cultivate the pleasure and simplicity of that kind of work; one day you'll miss it.

Responsibility

When you're old and eminent, what will you miss about being young and obscure? What people seem to miss most is the lack of responsibilities.

Responsibility is an occupational disease of eminence. In principle you could avoid it, just as in principle you could avoid getting fat as you get old, but few do. I sometimes suspect that responsibility is a trap and that the most virtuous route would be to shirk it, but regardless it's certainly constraining.

When you're an outsider you're constrained too, of course. You're short of money, for example. But that constrains you in different ways. How does responsibility constrain you? The worst thing is that it allows you not to focus on real work. Just as the most dangerous forms of procrastination are those that seem like work, the danger of responsibilities is not just that they can consume a whole day, but that they can do it without setting off the kind of alarms you'd set off if you spent a whole day sitting on a park bench.

A lot of the pain of being an outsider is being aware of one's own procrastination. But this is actually a good thing. You're at least close enough to work that the smell of it makes you hungry.

As an outsider, you're just one step away from getting things done. A huge step, admittedly, and one that most people never seem to make, but only one step. If you can summon up the energy to get started, you can work on projects with an intensity (in both senses) that few insiders can match. For insiders work turns into a duty, laden with responsibilities and expectations. It's never so pure as it was when they were young.

Work like a dog being taken for a walk, instead of an ox being yoked to the plow. That's what they miss.

Audience

A lot of outsiders make the mistake of doing the opposite; they admire the eminent so much that they copy even their flaws. Copying is a good way to learn, but copy the right things. When I was in college I imitated the pompous diction of famous professors. But this wasn't what *made* them eminent -- it was more a flaw their eminence had allowed them to sink into. Imitating it was like pretending to have gout in order to seem rich.

Half the distinguishing qualities of the eminent are actually disadvantages. Imitating these is not only a waste of time, but will make you seem a fool to your models, who are often well aware of it.

What are the genuine advantages of being an insider? The greatest is an audience. It often seems to outsiders that the great advantage of insiders is money -- that they have the resources to do what they want. But so do people who inherit money, and that doesn't seem to help, not as much as an audience. It's good for morale to know people want to see what you're making; it draws work out of you.

If I'm right that the defining advantage of insiders is an audience, then we live in exciting times, because just in the last ten years the Internet has made audiences a lot more liquid. Outsiders don't have to content themselves anymore with a proxy audience of a few smart friends. Now, thanks to the Internet, they can start to grow themselves actual audiences. This is great news for the marginal, who retain the advantages of outsiders while increasingly being able to siphon off what had till recently been the prerogative of the elite.

Though the Web has been around for more than ten years, I think we're just beginning to see its democratizing effects. Outsiders are still learning how to steal audiences. But more importantly, audiences are still learning how to be stolen -- they're still just beginning to realize how much deeper bloggers can dig than journalists, how much more interesting a democratic news site can be than a front page controlled by editors, and how much funnier a bunch of kids with webcams can be than mass-produced sitcoms.

The big media companies shouldn't worry that people will post their copyrighted material on YouTube. They should worry that people will post their own stuff on YouTube, and audiences will watch that instead.

Hacking

If I had to condense the power of the marginal into one sentence it would be: just try hacking something together. That phrase draws in most threads I've mentioned here. Hacking something together means deciding what to do as you're doing it, not a subordinate executing the vision of his boss. It implies the result won't be pretty, because it will be made quickly out of inadequate materials. It may work, but it won't be the sort of thing the eminent would want to put their name on. Something hacked together means something that barely solves the problem, or maybe doesn't solve the problem at all, but another you discovered en route. But that's ok, because the main value of that initial version is not the thing itself, but what it leads to. Insiders who daren't walk through the mud in their nice clothes will never make it to the solid ground on the other side.

The word "try" is an especially valuable component. I disagree here with Yoda, who said there is no try. There is try. It implies there's no punishment if you fail. You're driven by curiosity instead of duty. That means the wind of procrastination will be in your favor: instead of avoiding this work, this will be what you do as a way of avoiding other work. And when you do it, you'll be in a better mood. The more the work depends on imagination, the more that matters, because most people have more ideas when they're happy.

If I could go back and redo my twenties, that would be one thing I'd do more of: just try hacking things together. Like many people that age, I spent a lot of time worrying about what I should do. I also spent some time trying to build stuff. I should have spent less time worrying and more time building. If you're not sure what to do, make something.

Raymond Chandler's advice to thriller writers was "When in doubt, have a man come through a door with a gun in his hand." He followed that advice. Judging from his books, he was often in doubt. But though the result is occasionally cheesy, it's never boring. In life, as in books, action is underrated.

Fortunately the number of things you can just hack together keeps increasing. People fifty years ago would be astonished that one could just hack together a movie, for example. Now you can even hack together distribution. Just make stuff and put it online.

Inappropriate

If you really want to score big, the place to focus is the margin of the margin: the territories only recently captured from the insiders. That's where you'll find the juiciest projects still undone, either because they seemed too risky, or simply because there were too few insiders to explore everything.

This is why I spend most of my time writing essays lately. The writing of essays used to be limited to those who could get them published. In principle you could have written them and just shown them to your friends; in practice that didn't work. ⁶ An essayist needs the resistance of an audience, just as an engraver needs the resistance of the plate.

Up till a few years ago, writing essays was the ultimate insider's game. Domain experts were allowed to publish essays about their field, but the pool allowed to write on general topics was about eight people who went to the right parties in New York. Now the reconquista has overrun this territory, and, not surprisingly, found it sparsely cultivated. There are so many essays yet unwritten. They tend to be the naughtier ones; the insiders have pretty much exhausted the motherhood and apple pie topics.

This leads to my final suggestion: a technique for determining when you're on the right track. You're on the right track when people complain that you're unqualified, or that you've done something inappropriate. If people are complaining, that means you're doing something rather than sitting around, which is the first step. And if they're driven to such empty forms of complaint, that means you've probably done something good.

If you make something and people complain that it doesn't *work*, that's a problem. But if the worst thing they can hit you with is your own status as an outsider, that implies that in every other respect you've succeeded. Pointing out that someone is unqualified is as desperate as resorting to racial slurs. It's just a legitimate sounding way of saying: we don't like your type around here.

But the best thing of all is when people call what you're doing inappropriate. I've been hearing this word all my life and I only recently realized that it is, in fact, the sound of the homing beacon. "Inappropriate" is the null criticism. It's merely the adjective form of "I don't like it."

So that, I think, should be the highest goal for the marginal. Be inappropriate.

When you hear people saying that, you're golden. And they, incidentally, are busted.

Notes

[1] The facts about Apple's early history are from an interview with Steve Wozniak in Jessica Livingston's *Founders at Work*.

[2] As usual the popular image is several decades behind reality. Now the misunderstood artist is not a chain-smoking drunk who pours his soul into big, messy canvases that philistines see and say "that's not art" because it isn't a picture of anything. The philistines have now been trained that anything hung on a wall is art. Now the misunderstood artist is a coffee-drinking vegan cartoonist whose work they see and say "that's not art" because it looks like stuff they've seen in the Sunday paper.

[3] In fact this would do fairly well as a definition of politics: what determines rank in the absence of objective tests.

[4] In high school you're led to believe your whole future depends on where you go to college, but it turns out only to buy you a couple years. By your mid-twenties the people worth impressing already judge you more by what you've done than where you went to school.

[5] Managers are presumably wondering, how can I make this miracle happen? How can I make the people working for me do more with less? Unfortunately the constraint probably has to be self-imposed. If you're *expected* to do more with less, then you're being starved, not eating virtuously.

[6] Without the prospect of publication, the closest most people come to writing essays is to write in a journal. I find I never get as deeply into subjects as I do in proper essays. As the name implies, you don't go back and rewrite journal entries over and over for two weeks.

Thanks to Sam Altman, Trevor Blackwell, Paul Buchheit, Sarah Harlin, Jessica Livingston, Jackie McDonough, Robert Morris, Olin Shivers, and Chris Small for reading drafts of this, and to Chris Small and Chad Fowler for inviting me to speak.

Japanese Translation

Chinese Translation

069 The 18 Mistakes That Kill Startups

Want to start a startup? Get funded by Y Combinator.

October 2006

In the Q & A period after a recent talk, someone asked what made startups fail. After standing there gaping for a few seconds I realized this was kind of a trick question. It's equivalent to asking how to make a startup succeed -- if you avoid every cause of failure, you succeed -- and that's too big a question to answer on the fly.

Afterwards I realized it could be helpful to look at the problem from this direction. If you have a list of all the things you shouldn't do, you can turn that into a recipe for succeeding just by negating. And this form of list may be more useful in practice. It's easier to catch yourself doing something you shouldn't than always to remember to do something you should. 1

In a sense there's just one mistake that kills startups: not making something users want. If you make something users want, you'll probably be fine, whatever else you do or don't do. And if you don't make something users want, then you're dead, whatever else you do or don't do. So really this is a list of 18 things that cause startups not to make something users want. Nearly all failure funnels through that.

1. Single Founder

Have you ever noticed how few successful startups were founded by just one person? Even companies you think of as having one founder, like Oracle, usually turn out to have more. It seems unlikely this is a coincidence.

What's wrong with having one founder? To start with, it's a vote of no confidence. It probably means the founder couldn't talk any of his friends into starting the company with him. That's pretty alarming, because his friends are the ones who know him best.

But even if the founder's friends were all wrong and the company is a good bet, he's still at a disadvantage. Starting a startup is too hard for one person. Even if you could do all the work yourself, you need colleagues to brainstorm with, to talk you out of stupid decisions, and to cheer you up when things go wrong.

The last one might be the most important. The low points in a startup are so low that few could bear them alone. When you have multiple founders, esprit de corps binds them together in a way that seems to violate conservation laws. Each thinks "I can't let my friends down." This is one of the most powerful forces in human nature, and it's missing when there's just one founder.

2. Bad Location

Startups prosper in some places and not others. Silicon Valley dominates, then Boston, then Seattle, Austin, Denver, and New York. After that there's not much. Even in New York the number of startups per capita is probably a 20th of what it is in Silicon Valley. In towns like Houston and Chicago and Detroit it's too small to measure.

Why is the falloff so sharp? Probably for the same reason it is in other industries. What's the sixth largest fashion center in the US? The sixth largest center for

oil, or finance, or publishing? Whatever they are they're probably so far from the top that it would be misleading even to call them centers.

It's an interesting question why cities become startup hubs, but the reason startups prosper in them is probably the same as it is for any industry: that's where the experts are. Standards are higher; people are more sympathetic to what you're doing; the kind of people you want to hire want to live there; supporting industries are there; the people you run into in chance meetings are in the same business. Who knows exactly how these factors combine to boost startups in Silicon Valley and squish them in Detroit, but it's clear they do from the number of startups per capita in each.

3. Marginal Niche

Most of the groups that apply to Y Combinator suffer from a common problem: choosing a small, obscure niche in the hope of avoiding competition.

If you watch little kids playing sports, you notice that below a certain age they're afraid of the ball. When the ball comes near them their instinct is to avoid it. I didn't make a lot of catches as an eight year old outfielder, because whenever a fly ball came my way, I used to close my eyes and hold my glove up more for protection than in the hope of catching it.

Choosing a marginal project is the startup equivalent of my eight year old strategy for dealing with fly balls. If you make anything good, you're going to have competitors, so you may as well face that. You can only avoid competition by avoiding good ideas.

I think this shrinking from big problems is mostly unconscious. It's not that people think of grand ideas but decide to pursue smaller ones because they seem safer. Your unconscious won't even let you think of grand ideas. So the solution may be to think about ideas without involving yourself. What would be a great idea for *someone else* to do as a startup?

4. Derivative Idea

Many of the applications we get are imitations of some existing company. That's one source of ideas, but not the best. If you look at the origins of successful startups, few were started in imitation of some other startup. Where did they get their ideas? Usually from some specific, unsolved problem the founders identified.

Our startup made software for making online stores. When we started it, there wasn't any; the few sites you could order from were hand-made at great expense by web consultants. We knew that if online shopping ever took off, these sites would have to be generated by software, so we wrote some. Pretty straightforward.

It seems like the best problems to solve are ones that affect you personally. Apple happened because Steve Wozniak wanted a computer, Google because Larry

and Sergey couldn't find stuff online, Hotmail because Sabeer Bhatia and Jack Smith couldn't exchange email at work.

So instead of copying the Facebook, with some variation that the Facebook rightly ignored, look for ideas from the other direction. Instead of starting from companies and working back to the problems they solved, look for problems and imagine the company that might solve them. 2 What do people complain about? What do you wish there was?

5. Obstinacy

In some fields the way to succeed is to have a vision of what you want to achieve, and to hold true to it no matter what setbacks you encounter. Starting startups is not one of them. The stick-to-your-vision approach works for something like winning an Olympic gold medal, where the problem is well- defined. Startups are more like science, where you need to follow the trail wherever it leads.

So don't get too attached to your original plan, because it's probably wrong. Most successful startups end up doing something different than they originally intended -- often so different that it doesn't even seem like the same company. You have to be prepared to see the better idea when it arrives. And the hardest part of that is often discarding your old idea.

But openness to new ideas has to be tuned just right. Switching to a new idea every week will be equally fatal. Is there some kind of external test you can use? One is to ask whether the ideas represent some kind of progression. If in each new idea you're able to re-use most of what you built for the previous ones, then you're probably in a process that converges. Whereas if you keep restarting from scratch, that's a bad sign.

Fortunately there's someone you can ask for advice: your users. If you're thinking about turning in some new direction and your users seem excited about it, it's probably a good bet.

6. Hiring Bad Programmers

I forgot to include this in the early versions of the list, because nearly all the founders I know are programmers. This is not a serious problem for them. They might accidentally hire someone bad, but it's not going to kill the company. In a pinch they can do whatever's required themselves.

But when I think about what killed most of the startups in the e-commerce business back in the 90s, it was bad programmers. A lot of those companies were started by business guys who thought the way startups worked was that you had some clever idea and then hired programmers to implement it. That's actually much harder than it sounds -- almost impossibly hard in fact -- because business guys can't tell which are the good programmers. They don't even get a shot at the best ones, because no one really good wants a job implementing the vision of a business guy.

In practice what happens is that the business guys choose people they think are good programmers (it says here on his resume that he's a Microsoft Certified Developer) but who aren't. Then they're mystified to find that their startup lumbers along like a World War II bomber while their competitors scream past like jet fighters. This kind of startup is in the same position as a big company, but without the advantages.

So how do you pick good programmers if you're not a programmer? I don't think there's an answer. I was about to say you'd have to find a good programmer to help you hire people. But if you can't recognize good programmers, how would you even do that?

7. Choosing the Wrong Platform

A related problem (since it tends to be done by bad programmers) is choosing the wrong platform. For example, I think a lot of startups during the Bubble killed themselves by deciding to build server-based applications on Windows. Hotmail was still running on FreeBSD for years after Microsoft bought it, presumably because Windows couldn't handle the load. If Hotmail's founders had chosen to use Windows, they would have been swamped.

PayPal only just dodged this bullet. After they merged with X.com, the new CEO wanted to switch to Windows -- even after PayPal cofounder Max Levchin showed that their software scaled only 1% as well on Windows as Unix. Fortunately for PayPal they switched CEOs instead.

Platform is a vague word. It could mean an operating system, or a programming language, or a "framework" built on top of a programming language. It implies something that both supports and limits, like the foundation of a house.

The scary thing about platforms is that there are always some that seem to outsiders to be fine, responsible choices and yet, like Windows in the 90s, will destroy you if you choose them. Java applets were probably the most spectacular example. This was supposed to be the new way of delivering applications. Presumably it killed just about 100% of the startups who believed that.

How do you pick the right platforms? The usual way is to hire good programmers and let them choose. But there is a trick you could use if you're not a programmer: visit a top computer science department and see what they use in research projects.

8. Slowness in Launching

Companies of all sizes have a hard time getting software done. It's intrinsic to the medium; software is always 85% done. It takes an effort of will to push through this and get something released to users. 3

Startups make all kinds of excuses for delaying their launch. Most are equivalent to the ones people use for procrastinating in everyday life. There's something that needs to happen first. Maybe. But if the software were 100% finished and ready to launch at the push of a button, would they still be waiting?

One reason to launch quickly is that it forces you to actually *finish* some quantum of work. Nothing is truly finished till it's released; you can see that from the rush of work that's always involved in releasing anything, no matter how finished you thought it was. The other reason you need to launch is that it's only by bouncing your idea off users that you fully understand it.

Several distinct problems manifest themselves as delays in launching: working too slowly; not truly understanding the problem; fear of having to deal with users; fear of being judged; working on too many different things; excessive perfectionism. Fortunately you can combat all of them by the simple expedient of forcing yourself to launch *something* fairly quickly.

9. Launching Too Early

Launching too slowly has probably killed a hundred times more startups than launching too fast, but it is possible to launch too fast. The danger here is that you ruin your reputation. You launch something, the early adopters try it out, and if it's no good they may never come back.

So what's the minimum you need to launch? We suggest startups think about what they plan to do, identify a core that's both (a) useful on its own and (b) something that can be incrementally expanded into the whole project, and then get that done as soon as possible.

This is the same approach I (and many other programmers) use for writing software. Think about the overall goal, then start by writing the smallest subset of it that does anything useful. If it's a subset, you'll have to write it anyway, so in the worst case you won't be wasting your time. But more likely you'll find that implementing a working subset is both good for morale and helps you see more clearly what the rest should do.

The early adopters you need to impress are fairly tolerant. They don't expect a newly launched product to do everything; it just has to do *something*.

10. Having No Specific User in Mind

You can't build things users like without understanding them. I mentioned earlier that the most successful startups seem to have begun by trying to solve a problem their founders had. Perhaps there's a rule here: perhaps you create wealth in proportion to how well you understand the problem you're solving, and the problems you understand best are your own. ⁴

That's just a theory. What's not a theory is the converse: if you're trying to solve problems you don't understand, you're hosed.

And yet a surprising number of founders seem willing to assume that someone, they're not sure exactly who, will want what they're building. Do the founders want it? No, they're not the target market. Who is? Teenagers. People interested in local events (that one is a perennial tarpit). Or "business" users. What business users? Gas stations? Movie studios? Defense contractors?

You can of course build something for users other than yourself. We did. But you should realize you're stepping into dangerous territory. You're flying on instruments, in effect, so you should (a) consciously shift gears, instead of assuming you can rely on your intuitions as you ordinarily would, and (b) look at the instruments.

In this case the instruments are the users. When designing for other people you have to be empirical. You can no longer guess what will work; you have to find users and measure their responses. So if you're going to make something for teenagers or "business" users or some other group that doesn't include you, you have to be able to talk some specific ones into using what you're making. If you can't, you're on the wrong track.

11. Raising Too Little Money

Most successful startups take funding at some point. Like having more than one founder, it seems a good bet statistically. How much should you take, though?

Startup funding is measured in time. Every startup that isn't profitable (meaning nearly all of them, initially) has a certain amount of time left before the money runs out and they have to stop. This is sometimes referred to as runway, as in "How much runway do you have left?" It's a good metaphor because it reminds you that when the money runs out you're going to be airborne or dead.

Too little money means not enough to get airborne. What airborne means depends on the situation. Usually you have to advance to a visibly higher level: if all you have is an idea, a working prototype; if you have a prototype, launching; if you're launched, significant growth. It depends on investors, because until you're profitable that's who you have to convince.

So if you take money from investors, you have to take enough to get to the next step, whatever that is. 5 Fortunately you have some control over both how much you spend and what the next step is. We advise startups to set both low, initially: spend practically nothing, and make your initial goal simply to build a solid prototype. This gives you maximum flexibility.

12. Spending Too Much

It's hard to distinguish spending too much from raising too little. If you run out of money, you could say either was the cause. The only way to decide which to call it is by comparison with other startups. If you raised five million and ran out of money, you probably spent too much.

Burning through too much money is not as common as it used to be. Founders seem to have learned that lesson. Plus it keeps getting cheaper to start a startup. So as of this writing few startups spend too much. None of the ones we've funded have. (And not just because we make small investments; many have gone on to raise further rounds.)

The classic way to burn through cash is by hiring a lot of people. This bites you twice: in addition to increasing your costs, it slows you down--so money that's

getting consumed faster has to last longer. Most hackers understand why that happens; Fred Brooks explained it in *The Mythical Man-Month*.

We have three general suggestions about hiring: (a) don't do it if you can avoid it, (b) pay people with equity rather than salary, not just to save money, but because you want the kind of people who are committed enough to prefer that, and (c) only hire people who are either going to write code or go out and get users, because those are the only things you need at first.

13. Raising Too Much Money

It's obvious how too little money could kill you, but is there such a thing as having too much?

Yes and no. The problem is not so much the money itself as what comes with it. As one VC who spoke at Y Combinator said, "Once you take several million dollars of my money, the clock is ticking." If VCs fund you, they're not going to let you just put the money in the bank and keep operating as two guys living on ramen. They want that money to go to work. ⁶ At the very least you'll move into proper office space and hire more people. That will change the atmosphere, and not entirely for the better. Now most of your people will be employees rather than founders. They won't be as committed; they'll need to be told what to do; they'll start to engage in office politics.

When you raise a lot of money, your company moves to the suburbs and has kids.

Perhaps more dangerously, once you take a lot of money it gets harder to change direction. Suppose your initial plan was to sell something to companies. After taking VC money you hire a sales force to do that. What happens now if you realize you should be making this for consumers instead of businesses? That's a completely different kind of selling. What happens, in practice, is that you don't realize that. The more people you have, the more you stay pointed in the same direction.

Another drawback of large investments is the time they take. The time required to raise money grows with the amount. ⁷ When the amount rises into the millions, investors get very cautious. VCs never quite say yes or no; they just engage you in an apparently endless conversation. Raising VC scale investments is thus a huge time sink -- more work, probably, than the startup itself. And you don't want to be spending all your time talking to investors while your competitors are spending theirs building things.

We advise founders who go on to seek VC money to take the first reasonable deal they get. If you get an offer from a reputable firm at a reasonable valuation with no unusually onerous terms, just take it and get on with building the company. ⁸ Who cares if you could get a 30% better deal elsewhere? Economically, startups are an all-or-nothing game. Bargain-hunting among investors is a waste of time.

14. Poor Investor Management

As a founder, you have to manage your investors. You shouldn't ignore them, because they may have useful insights. But neither should you let them run the company. That's supposed to be your job. If investors had sufficient vision to run the companies they fund, why didn't they start them?

Pissing off investors by ignoring them is probably less dangerous than caving in to them. In our startup, we erred on the ignoring side. A lot of our energy got drained away in disputes with investors instead of going into the product. But this was less costly than giving in, which would probably have destroyed the company. If the founders know what they're doing, it's better to have half their attention focused on the product than the full attention of investors who don't.

How hard you have to work on managing investors usually depends on how much money you've taken. When you raise VC-scale money, the investors get a great deal of control. If they have a board majority, they're literally your bosses. In the more common case, where founders and investors are equally represented and the deciding vote is cast by neutral outside directors, all the investors have to do is convince the outside directors and they control the company.

If things go well, this shouldn't matter. So long as you seem to be advancing rapidly, most investors will leave you alone. But things don't always go smoothly in startups. Investors have made trouble even for the most successful companies. One of the most famous examples is Apple, whose board made a nearly fatal blunder in firing Steve Jobs. Apparently even Google got a lot of grief from their investors early on.

15. Sacrificing Users to (Supposed) Profit

When I said at the beginning that if you make something users want, you'll be fine, you may have noticed I didn't mention anything about having the right business model. That's not because making money is unimportant. I'm not suggesting that founders start companies with no chance of making money in the hope of unloading them before they tank. The reason we tell founders not to worry about the business model initially is that making something people want is so much harder.

I don't know why it's so hard to make something people want. It seems like it should be straightforward. But you can tell it must be hard by how few startups do it.

Because making something people want is so much harder than making money from it, you should leave business models for later, just as you'd leave some trivial but messy feature for version 2. In version 1, solve the core problem. And the core problem in a startup is how to create wealth (= how much people want something x the number who want it), not how to convert that wealth into money.

The companies that win are the ones that put users first. Google, for example. They made search work, then worried about how to make money from it. And yet some startup founders still think it's irresponsible not to focus on the business

model from the beginning. They're often encouraged in this by investors whose experience comes from less malleable industries.

It *is* irresponsible not to think about business models. It's just ten times more irresponsible not to think about the product.

16. Not Wanting to Get Your Hands Dirty

Nearly all programmers would rather spend their time writing code and have someone else handle the messy business of extracting money from it. And not just the lazy ones. Larry and Sergey apparently felt this way too at first. After developing their new search algorithm, the first thing they tried was to get some other company to buy it.

Start a company? Yech. Most hackers would rather just have ideas. But as Larry and Sergey found, there's not much of a market for ideas. No one trusts an idea till you embody it in a product and use that to grow a user base. Then they'll pay big time.

Maybe this will change, but I doubt it will change much. There's nothing like users for convincing acquirers. It's not just that the risk is decreased. The acquirers are human, and they have a hard time paying a bunch of young guys millions of dollars just for being clever. When the idea is embodied in a company with a lot of users, they can tell themselves they're buying the users rather than the cleverness, and this is easier for them to swallow. 9

If you're going to attract users, you'll probably have to get up from your computer and go find some. It's unpleasant work, but if you can make yourself do it you have a much greater chance of succeeding. In the first batch of startups we funded, in the summer of 2005, most of the founders spent all their time building their applications. But there was one who was away half the time talking to executives at cell phone companies, trying to arrange deals. Can you imagine anything more painful for a hacker? 10 But it paid off, because this startup seems the most successful of that group by an order of magnitude.

If you want to start a startup, you have to face the fact that you can't just hack. At least one hacker will have to spend some of the time doing business stuff.

17. Fights Between Founders

Fights between founders are surprisingly common. About 20% of the startups we've funded have had a founder leave. It happens so often that we've reversed our attitude to vesting. We still don't require it, but now we advise founders to vest so there will be an orderly way for people to quit.

A founder leaving doesn't necessarily kill a startup, though. Plenty of successful startups have had that happen. 11 Fortunately it's usually the least committed founder who leaves. If there are three founders and one who was lukewarm leaves, big deal. If you have two and one leaves, or a guy with critical technical skills leaves, that's more of a problem. But even that is survivable. Blogger got down to one person, and they bounced back.

Most of the disputes I've seen between founders could have been avoided if they'd been more careful about who they started a company with. Most disputes are not due to the situation but the people. Which means they're inevitable. And most founders who've been burned by such disputes probably had misgivings, which they suppressed, when they started the company. Don't suppress misgivings. It's much easier to fix problems before the company is started than after. So don't include your housemate in your startup because he'd feel left out otherwise. Don't start a company with someone you dislike because they have some skill you need and you worry you won't find anyone else. The people are the most important ingredient in a startup, so don't compromise there.

18. A Half-Hearted Effort

The failed startups you hear most about are the spectacular flameouts. Those are actually the elite of failures. The most common type is not the one that makes spectacular mistakes, but the one that doesn't do much of anything -- the one we never even hear about, because it was some project a couple guys started on the side while working on their day jobs, but which never got anywhere and was gradually abandoned.

Statistically, if you want to avoid failure, it would seem like the most important thing is to quit your day job. Most founders of failed startups don't quit their day jobs, and most founders of successful ones do. If startup failure were a disease, the CDC would be issuing bulletins warning people to avoid day jobs.

Does that mean you should quit your day job? Not necessarily. I'm guessing here, but I'd guess that many of these would-be founders may not have the kind of determination it takes to start a company, and that in the back of their minds, they know it. The reason they don't invest more time in their startup is that they know it's a bad investment. ¹²

I'd also guess there's some band of people who could have succeeded if they'd taken the leap and done it full-time, but didn't. I have no idea how wide this band is, but if the winner/borderline/hopeless progression has the sort of distribution you'd expect, the number of people who could have made it, if they'd quit their day job, is probably an order of magnitude larger than the number who do make it. ¹³

If that's true, most startups that could succeed fail because the founders don't devote their whole efforts to them. That certainly accords with what I see out in the world. Most startups fail because they don't make something people want, and the reason most don't is that they don't try hard enough.

In other words, starting startups is just like everything else. The biggest mistake you can make is not to try hard enough. To the extent there's a secret to success, it's not to be in denial about that.

Notes

[1] This is not a complete list of the causes of failure, just those you can control.

There are also several you can't, notably ineptitude and bad luck.

[2] Ironically, one variant of the Facebook that might work is a facebook exclusively for college students.

[3] Steve Jobs tried to motivate people by saying "Real artists ship." This is a fine sentence, but unfortunately not true. Many famous works of art are unfinished. It's true in fields that have hard deadlines, like architecture and filmmaking, but even there people tend to be tweaking stuff till it's yanked out of their hands.

[4] There's probably also a second factor: startup founders tend to be at the leading edge of technology, so problems they face are probably especially valuable.

[5] You should take more than you think you'll need, maybe 50% to 100% more, because software takes longer to write and deals longer to close than you expect.

[6] Since people sometimes call us VCs, I should add that we're not. VCs invest large amounts of other people's money. We invest small amounts of our own, like angel investors.

[7] Not linearly of course, or it would take forever to raise five million dollars. In practice it just feels like it takes forever.

Though if you include the cases where VCs don't invest, it would literally take forever in the median case. And maybe we should, because the danger of chasing large investments is not just that they take a long time. That's the *best* case. The real danger is that you'll expend a lot of time and get nothing.

[8] Some VCs will offer you an artificially low valuation to see if you have the balls to ask for more. It's lame that VCs play such games, but some do. If you're dealing with one of those you should push back on the valuation a bit.

[9] Suppose YouTube's founders had gone to Google in 2005 and told them "Google Video is badly designed. Give us \$10 million and we'll tell you all the mistakes you made." They would have gotten the royal raspberry. Eighteen months later Google paid \$1.6 billion for the same lesson, partly because they could then tell themselves that they were buying a phenomenon, or a community, or some vague thing like that.

I don't mean to be hard on Google. They did better than their competitors, who may have now missed the video boat entirely.

[10] Yes, actually: dealing with the government. But phone companies are up there.

[11] Many more than most people realize, because companies don't advertise this. Did you know Apple originally had three founders?

[12] I'm not dissing these people. I don't have the determination myself. I've twice come close to starting startups since Viaweb, and both times I bailed because I realized that without the spur of poverty I just wasn't willing to endure the stress of a startup.

[13] So how do you know whether you're in the category of people who should quit their day job, or the presumably larger one who shouldn't? I got to the point of saying that this was hard to judge for yourself and that you should seek outside advice, before realizing that that's what we do. We think of ourselves as investors, but viewed from the other direction Y Combinator is a service for advising people whether or not to quit their day job. We could be mistaken, and no doubt often are, but we do at least bet money on our conclusions.

Thanks to Sam Altman, Jessica Livingston, Greg McAdoo, and Robert Morris for reading drafts of this.

Japanese Translation

Spanish Translation

Romanian Translation

Chinese Translation

Arabic Translation

089 You Weren't Meant to Have a Boss

Want to start a startup? Get funded by Y Combinator.

March 2008, rev. June 2008

Technology tends to separate normal from natural. Our bodies weren't designed to eat the foods that people in rich countries eat, or to get so little exercise. There may be a similar problem with the way we work: a normal job may be as bad for us intellectually as white flour or sugar is for us physically.

I began to suspect this after spending several years working with startup founders. I've now worked with over 200 of them, and I've noticed a definite difference between programmers working on their own startups and those working for large organizations. I wouldn't say founders seem happier, necessarily; starting a startup can be very stressful. Maybe the best way to put it is to say that they're happier in the sense that your body is happier during a long run than sitting on a sofa eating doughnuts.

Though they're statistically abnormal, startup founders seem to be working in a way that's more natural for humans.

I was in Africa last year and saw a lot of animals in the wild that I'd only seen in zoos before. It was remarkable how different they seemed. Particularly lions. Lions in the wild seem about ten times more alive. They're like different animals. I suspect that working for oneself feels better to humans in much the same way that living in the wild must feel better to a wide-ranging predator like a lion. Life in a zoo is easier, but it isn't the life they were designed for.

Trees

What's so unnatural about working for a big company? The root of the problem is that humans weren't meant to work in such large groups.

Another thing you notice when you see animals in the wild is that each species thrives in groups of a certain size. A herd of impalas might have 100 adults; baboons maybe 20; lions rarely 10. Humans also seem designed to work in groups, and what I've read about hunter-gatherers accords with research on organizations and my own experience to suggest roughly what the ideal size is: groups of 8 work well; by 20 they're getting hard to manage; and a group of 50 is really unwieldy. ¹

Whatever the upper limit is, we are clearly not meant to work in groups of several hundred. And yet--for reasons having more to do with technology than human nature--a great many people work for companies with hundreds or thousands of employees.

Companies know groups that large wouldn't work, so they divide themselves into units small enough to work together. But to coordinate these they have to introduce something new: bosses.

These smaller groups are always arranged in a tree structure. Your boss is the point where your group attaches to the tree. But when you use this trick for dividing a large group into smaller ones, something strange happens that I've never heard anyone mention explicitly. In the group one level up from yours, your boss represents your entire group. A group of 10 managers is not merely a group of 10 people working together in the usual way. It's really a group of groups. Which means for a group of 10 managers to work together as if they were simply a group of 10 individuals, the group working for each manager would have to work as if they were a single person--the workers and manager would each share only one person's worth of freedom between them.

In practice a group of people are never able to act as if they were one person. But in a large organization divided into groups in this way, the pressure is always in that direction. Each group tries its best to work as if it were the small group of individuals that humans were designed to work in. That was the point of creating it. And when you propagate that constraint, the result is that each person gets freedom of action in inverse proportion to the size of the entire tree. ²

Anyone who's worked for a large organization has felt this. You can feel the difference between working for a company with 100 employees and one with 10,000, even if your group has only 10 people.

Corn Syrup

A group of 10 people within a large organization is a kind of fake tribe. The number of people you interact with is about right. But something is missing: individual initiative. Tribes of hunter-gatherers have much more freedom. The

leaders have a little more power than other members of the tribe, but they don't generally tell them what to do and when the way a boss can.

It's not your boss's fault. The real problem is that in the group above you in the hierarchy, your entire group is one virtual person. Your boss is just the way that constraint is imparted to you.

So working in a group of 10 people within a large organization feels both right and wrong at the same time. On the surface it feels like the kind of group you're meant to work in, but something major is missing. A job at a big company is like high fructose corn syrup: it has some of the qualities of things you're meant to like, but is disastrously lacking in others.

Indeed, food is an excellent metaphor to explain what's wrong with the usual sort of job.

For example, working for a big company is the default thing to do, at least for programmers. How bad could it be? Well, food shows that pretty clearly. If you were dropped at a random point in America today, nearly all the food around you would be bad for you. Humans were not designed to eat white flour, refined sugar, high fructose corn syrup, and hydrogenated vegetable oil. And yet if you analyzed the contents of the average grocery store you'd probably find these four ingredients accounted for most of the calories. "Normal" food is terribly bad for you. The only people who eat what humans were actually designed to eat are a few Birkenstock-wearing weirdos in Berkeley.

If "normal" food is so bad for us, why is it so common? There are two main reasons. One is that it has more immediate appeal. You may feel lousy an hour after eating that pizza, but eating the first couple bites feels great. The other is economies of scale. Producing junk food scales; producing fresh vegetables doesn't. Which means (a) junk food can be very cheap, and (b) it's worth spending a lot to market it.

If people have to choose between something that's cheap, heavily marketed, and appealing in the short term, and something that's expensive, obscure, and appealing in the long term, which do you think most will choose?

It's the same with work. The average MIT graduate wants to work at Google or Microsoft, because it's a recognized brand, it's safe, and they'll get paid a good salary right away. It's the job equivalent of the pizza they had for lunch. The drawbacks will only become apparent later, and then only in a vague sense of malaise.

And founders and early employees of startups, meanwhile, are like the Birkenstock-wearing weirdos of Berkeley: though a tiny minority of the population, they're the ones living as humans are meant to. In an artificial world, only extremists live naturally.

Programmers

The restrictiveness of big company jobs is particularly hard on programmers, because the essence of programming is to build new things. Sales people make much the same pitches every day; support people answer much the same questions; but once you've written a piece of code you don't need to write it again. So a programmer working as programmers are meant to is always making new things. And when you're part of an organization whose structure gives each person freedom in inverse proportion to the size of the tree, you're going to face resistance when you do something new.

This seems an inevitable consequence of bigness. It's true even in the smartest companies. I was talking recently to a founder who considered starting a startup right out of college, but went to work for Google instead because he thought he'd learn more there. He didn't learn as much as he expected. Programmers learn by doing, and most of the things he wanted to do, he couldn't--sometimes because the company wouldn't let him, but often because the company's code wouldn't let him. Between the drag of legacy code, the overhead of doing development in such a large organization, and the restrictions imposed by interfaces owned by other groups, he could only try a fraction of the things he would have liked to. He said he has learned much more in his own startup, despite the fact that he has to do all the company's errands as well as programming, because at least when he's programming he can do whatever he wants.

An obstacle downstream propagates upstream. If you're not allowed to implement new ideas, you stop having them. And vice versa: when you can do whatever you want, you have more ideas about what to do. So working for yourself makes your brain more powerful in the same way a low-restriction exhaust system makes an engine more powerful.

Working for yourself doesn't have to mean starting a startup, of course. But a programmer deciding between a regular job at a big company and their own startup is probably going to learn more doing the startup.

You can adjust the amount of freedom you get by scaling the size of company you work for. If you start the company, you'll have the most freedom. If you become one of the first 10 employees you'll have almost as much freedom as the founders. Even a company with 100 people will feel different from one with 1000.

Working for a small company doesn't ensure freedom. The tree structure of large organizations sets an upper bound on freedom, not a lower bound. The head of a small company may still choose to be a tyrant. The point is that a large organization is compelled by its structure to be one.

Consequences

That has real consequences for both organizations and individuals. One is that companies will inevitably slow down as they grow larger, no matter how hard they try to keep their startup mojo. It's a consequence of the tree structure that every large organization is forced to adopt.

Or rather, a large organization could only avoid slowing down if they avoided

tree structure. And since human nature limits the size of group that can work together, the only way I can imagine for larger groups to avoid tree structure would be to have no structure: to have each group actually be independent, and to work together the way components of a market economy do.

That might be worth exploring. I suspect there are already some highly partitionable businesses that lean this way. But I don't know any technology companies that have done it.

There is one thing companies can do short of structuring themselves as sponges: they can stay small. If I'm right, then it really pays to keep a company as small as it can be at every stage. Particularly a technology company. Which means it's doubly important to hire the best people. Mediocre hires hurt you twice: they get less done, but they also make you big, because you need more of them to solve a given problem.

For individuals the upshot is the same: aim small. It will always suck to work for large organizations, and the larger the organization, the more it will suck.

In an essay I wrote a couple years ago I advised graduating seniors to work for a couple years for another company before starting their own. I'd modify that now. Work for another company if you want to, but only for a small one, and if you want to start your own startup, go ahead.

The reason I suggested college graduates not start startups immediately was that I felt most would fail. And they will. But ambitious programmers are better off doing their own thing and failing than going to work at a big company. Certainly they'll learn more. They might even be better off financially. A lot of people in their early twenties get into debt, because their expenses grow even faster than the salary that seemed so high when they left school. At least if you start a startup and fail your net worth will be zero rather than negative. ³

We've now funded so many different types of founders that we have enough data to see patterns, and there seems to be no benefit from working for a big company. The people who've worked for a few years do seem better than the ones straight out of college, but only because they're that much older.

The people who come to us from big companies often seem kind of conservative. It's hard to say how much is because big companies made them that way, and how much is the natural conservatism that made them work for the big companies in the first place. But certainly a large part of it is learned. I know because I've seen it burn off.

Having seen that happen so many times is one of the things that convinces me that working for oneself, or at least for a small group, is the natural way for programmers to live. Founders arriving at Y Combinator often have the downtrodden air of refugees. Three months later they're transformed: they have so much more confidence that they seem as if they've grown several inches taller. ⁴ Strange as this sounds, they seem both more worried and happier at the same time. Which is exactly how I'd describe the way lions seem in the wild.

Watching employees get transformed into founders makes it clear that the difference between the two is due mostly to environment--and in particular that the environment in big companies is toxic to programmers. In the first couple weeks of working on their own startup they seem to come to life, because finally they're working the way people are meant to.

Notes

[1] When I talk about humans being meant or designed to live a certain way, I mean by evolution.

[2] It's not only the leaves who suffer. The constraint propagates up as well as down. So managers are constrained too; instead of just doing things, they have to act through subordinates.

[3] Do not finance your startup with credit cards. Financing a startup with debt is usually a stupid move, and credit card debt stupidest of all. Credit card debt is a bad idea, period. It is a trap set by evil companies for the desperate and the foolish.

[4] The founders we fund used to be younger (initially we encouraged undergrads to apply), and the first couple times I saw this I used to wonder if they were actually getting physically taller.

Thanks to Trevor Blackwell, Ross Boucher, Aaron Iba, Abby Kirigin, Ivan Kirigin, Jessica Livingston, and Robert Morris for reading drafts of this.

French Translation

Russian Translation

096 Cities and Ambition

May 2008

Great cities attract ambitious people. You can sense it when you walk around one. In a hundred subtle ways, the city sends you a message: you could do more; you should try harder.

The surprising thing is how different these messages can be. New York tells you, above all: you should make more money. There are other messages too, of course. You should be hipper. You should be better looking. But the clearest message is that you should be richer.

What I like about Boston (or rather Cambridge) is that the message there is: you should be smarter. You really should get around to reading all those books you've been meaning to.

When you ask what message a city sends, you sometimes get surprising answers. As much as they respect brains in Silicon Valley, the message the Valley sends is: you should be more powerful.

That's not quite the same message New York sends. Power matters in New York too of course, but New York is pretty impressed by a billion dollars even if you merely inherited it. In Silicon Valley no one would care except a few real estate agents. What matters in Silicon Valley is how much effect you have on the world. The reason people there care about Larry and Sergey is not their wealth but the fact that they control Google, which affects practically everyone.

How much does it matter what message a city sends? Empirically, the answer seems to be: a lot. You might think that if you had enough strength of mind to do great things, you'd be able to transcend your environment. Where you live should make at most a couple percent difference. But if you look at the historical evidence, it seems to matter more than that. Most people who did great things were clumped together in a few places where that sort of thing was done at the time.

You can see how powerful cities are from something I wrote about earlier: the case of the Milanese Leonardo. Practically every fifteenth century Italian painter you've heard of was from Florence, even though Milan was just as big. People in Florence weren't genetically different, so you have to assume there was someone born in Milan with as much natural ability as Leonardo. What happened to him?

If even someone with the same natural ability as Leonardo couldn't beat the force of environment, do you suppose you can?

I don't. I'm fairly stubborn, but I wouldn't try to fight this force. I'd rather use it. So I've thought a lot about where to live.

I'd always imagined Berkeley would be the ideal place -- that it would basically be Cambridge with good weather. But when I finally tried living there a couple years ago, it turned out not to be. The message Berkeley sends is: you should live better. Life in Berkeley is very civilized. It's probably the place in America where someone from Northern Europe would feel most at home. But it's not humming with ambition.

In retrospect it shouldn't have been surprising that a place so pleasant would attract people interested above all in quality of life. Cambridge with good weather, it turns out, is not Cambridge. The people you find in Cambridge are not there by accident. You have to make sacrifices to live there. It's expensive and somewhat grubby, and the weather's often bad. So the kind of people you find in Cambridge are the kind of people who want to live where the smartest people are, even if that means living in an expensive, grubby place with bad weather.

As of this writing, Cambridge seems to be the intellectual capital of the world. I realize that seems a preposterous claim. What makes it true is that it's more preposterous to claim about anywhere else. American universities currently seem to be the best, judging from the flow of ambitious students. And what US city has a stronger claim? New York? A fair number of smart people, but diluted by a much larger number of neanderthals in suits. The Bay Area has a lot of smart people too, but again, diluted; there are two great universities, but they're far apart. Harvard and MIT are practically adjacent by West Coast standards, and they're surrounded by about 20 other colleges and universities. 1

Cambridge as a result feels like a town whose main industry is ideas, while New York's is finance and Silicon Valley's is startups.

When you talk about cities in the sense we are, what you're really talking about is collections of people. For a long time cities were the only large collections of people, so you could use the two ideas interchangeably. But we can see how much things are changing from the examples I've mentioned. New York is a classic great city. But Cambridge is just part of a city, and Silicon Valley is not even that. (San Jose is not, as it sometimes claims, the capital of Silicon Valley. It's just 178 square miles at one end of it.)

Maybe the Internet will change things further. Maybe one day the most important community you belong to will be a virtual one, and it won't matter where you live physically. But I wouldn't bet on it. The physical world is very high bandwidth, and some of the ways cities send you messages are quite subtle.

One of the exhilarating things about coming back to Cambridge every spring is walking through the streets at dusk, when you can see into the houses. When you walk through Palo Alto in the evening, you see nothing but the blue glow of TVs. In Cambridge you see shelves full of promising-looking books. Palo Alto was probably much like Cambridge in 1960, but you'd never guess now that there was a university nearby. Now it's just one of the richer neighborhoods in Silicon Valley. 2

A city speaks to you mostly by accident -- in things you see through windows, in conversations you overhear. It's not something you have to seek out, but something you can't turn off. One of the occupational hazards of living in Cambridge is overhearing the conversations of people who use interrogative intonation in declarative sentences. But on average I'll take Cambridge conversations over New York or Silicon Valley ones.

A friend who moved to Silicon Valley in the late 90s said the worst thing about living there was the low quality of the eavesdropping. At the time I thought she was being deliberately eccentric. Sure, it can be interesting to eavesdrop on people, but is good quality eavesdropping so important that it would affect where you chose to live? Now I understand what she meant. The conversations you overhear tell you what sort of people you're among.

No matter how determined you are, it's hard not to be influenced by the people around you. It's not so much that you do whatever a city expects of you, but that you get discouraged when no one around you cares about the same things you do.

There's an imbalance between encouragement and discouragement like that between gaining and losing money. Most people overvalue negative amounts of money: they'll work much harder to avoid losing a dollar than to gain one. Similarly, although there are plenty of people strong enough to resist doing something just because that's what one is supposed to do where they happen to be, there are few strong enough to keep working on something no one around them cares about.

Because ambitions are to some extent incompatible and admiration is a zero-sum game, each city tends to focus on one type of ambition. The reason Cambridge is the intellectual capital is not just that there's a concentration of smart people there, but that there's nothing *else* people there care about more. Professors in New York and the Bay area are second class citizens -- till they start hedge funds or startups respectively.

This suggests an answer to a question people in New York have wondered about since the Bubble: whether New York could grow into a startup hub to rival Silicon Valley. One reason that's unlikely is that someone starting a startup in New York would feel like a second class citizen. ³ There's already something else people in New York admire more.

In the long term, that could be a bad thing for New York. The power of an important new technology does eventually convert to money. So by caring more about money and less about power than Silicon Valley, New York is recognizing the same thing, but slower. ⁴ And in fact it has been losing to Silicon Valley at its own game: the ratio of New York to California residents in the Forbes 400 has decreased from 1.45 (81:56) when the list was first published in 1982 to .83 (73:88) in 2007.

Not all cities send a message. Only those that are centers for some type of ambition do. And it can be hard to tell exactly what message a city sends without living there. I understand the messages of New York, Cambridge, and Silicon Valley because I've lived for several years in each of them. DC and LA seem to send messages too, but I haven't spent long enough in either to say for sure what they are.

The big thing in LA seems to be fame. There's an A List of people who are most in demand right now, and what's most admired is to be on it, or friends with those who are. Beneath that, the message is much like New York's, though perhaps with more emphasis on physical attractiveness.

In DC the message seems to be that the most important thing is who you know. You want to be an insider. In practice this seems to work much as in LA. There's an A List and you want to be on it or close to those who are. The only difference is how the A List is selected. And even that is not that different.

At the moment, San Francisco's message seems to be the same as Berkeley's: you should live better. But this will change if enough startups choose SF over the Valley. During the Bubble that was a predictor of failure -- a self-indulgent choice, like buying expensive office furniture. Even now I'm suspicious when startups choose SF. But if enough good ones do, it stops being a self-indulgent choice, because the center of gravity of Silicon Valley will shift there.

I haven't found anything like Cambridge for intellectual ambition. Oxford and Cambridge (England) feel like Ithaca or Hanover: the message is there, but not as strong.

Paris was once a great intellectual center. If you went there in 1300, it might have sent the message Cambridge does now. But I tried living there for a bit last year, and the ambitions of the inhabitants are not intellectual ones. The message Paris sends now is: do things with style. I liked that, actually. Paris is the only city I've lived in where people genuinely cared about art. In America only a few rich people buy original art, and even the more sophisticated ones rarely get past judging it by the brand name of the artist. But looking through windows at dusk in Paris you can see that people there actually care what paintings look like. Visually, Paris has the best eavesdropping I know. 5

There's one more message I've heard from cities: in London you can still (barely) hear the message that one should be more aristocratic. If you listen for it you can also hear it in Paris, New York, and Boston. But this message is everywhere very faint. It would have been strong 100 years ago, but now I probably wouldn't have picked it up at all if I hadn't deliberately tuned in to that wavelength to see if there was any signal left.

So far the complete list of messages I've picked up from cities is: wealth, style, hipness, physical attractiveness, fame, political power, economic power, intelligence, social class, and quality of life.

My immediate reaction to this list is that it makes me slightly queasy. I'd always considered ambition a good thing, but I realize now that was because I'd always implicitly understood it to mean ambition in the areas I cared about. When you list everything ambitious people are ambitious about, it's not so pretty.

On closer examination I see a couple things on the list that are surprising in the light of history. For example, physical attractiveness wouldn't have been there 100 years ago (though it might have been 2400 years ago). It has always mattered for women, but in the late twentieth century it seems to have started to matter for men as well. I'm not sure why -- probably some combination of the increasing power of women, the increasing influence of actors as models, and

the fact that so many people work in offices now: you can't show off by wearing clothes too fancy to wear in a factory, so you have to show off with your body instead.

Hipness is another thing you wouldn't have seen on the list 100 years ago. Or wouldn't you? What it means is to know what's what. So maybe it has simply replaced the component of social class that consisted of being "au fait." That could explain why hipness seems particularly admired in London: it's version 2 of the traditional English delight in obscure codes that only insiders understand.

Economic power would have been on the list 100 years ago, but what we mean by it is changing. It used to mean the control of vast human and material resources. But increasingly it means the ability to direct the course of technology, and some of the people in a position to do that are not even rich -- leaders of important open source projects, for example. The Captains of Industry of times past had laboratories full of clever people cooking up new technologies for them. The new breed are themselves those people.

As this force gets more attention, another is dropping off the list: social class. I think the two changes are related. Economic power, wealth, and social class are just names for the same thing at different stages in its life: economic power converts to wealth, and wealth to social class. So the focus of admiration is simply shifting upstream.

Does anyone who wants to do great work have to live in a great city? No; all great cities inspire some sort of ambition, but they aren't the only places that do. For some kinds of work, all you need is a handful of talented colleagues.

What cities provide is an audience, and a funnel for peers. These aren't so critical in something like math or physics, where no audience matters except your peers, and judging ability is sufficiently straightforward that hiring and admissions committees can do it reliably. In a field like math or physics all you need is a department with the right colleagues in it. It could be anywhere -- in Los Alamos, New Mexico, for example.

It's in fields like the arts or writing or technology that the larger environment matters. In these the best practitioners aren't conveniently collected in a few top university departments and research labs -- partly because talent is harder to judge, and partly because people pay for these things, so one doesn't need to rely on teaching or research funding to support oneself. It's in these more chaotic fields that it helps most to be in a great city: you need the encouragement of feeling that people around you care about the kind of work you do, and since you have to find peers for yourself, you need the much larger intake mechanism of a great city.

You don't have to live in a great city your whole life to benefit from it. The critical years seem to be the early and middle ones of your career. Clearly you don't have to grow up in a great city. Nor does it seem to matter if you go to

college in one. To most college students a world of a few thousand people seems big enough. Plus in college you don't yet have to face the hardest kind of work -- discovering new problems to solve.

It's when you move on to the next and much harder step that it helps most to be in a place where you can find peers and encouragement. You seem to be able to leave, if you want, once you've found both. The Impressionists show the typical pattern: they were born all over France (Pissarro was born in the Carribbean) and died all over France, but what defined them were the years they spent together in Paris.

Unless you're sure what you want to do and where the leading center for it is, your best bet is probably to try living in several places when you're young. You can never tell what message a city sends till you live there, or even whether it still sends one. Often your information will be wrong: I tried living in Florence when I was 25, thinking it would be an art center, but it turned out I was 450 years too late.

Even when a city is still a live center of ambition, you won't know for sure whether its message will resonate with you till you hear it. When I moved to New York, I was very excited at first. It's an exciting place. So it took me quite a while to realize I just wasn't like the people there. I kept searching for the Cambridge of New York. It turned out it was way, way uptown: an hour uptown by air.

Some people know at 16 what sort of work they're going to do, but in most ambitious kids, ambition seems to precede anything specific to be ambitious about. They know they want to do something great. They just haven't decided yet whether they're going to be a rock star or a brain surgeon. There's nothing wrong with that. But it means if you have this most common type of ambition, you'll probably have to figure out where to live by trial and error. You'll probably have to find the city where you feel at home to know what sort of ambition you have.

Notes

[1] This is one of the advantages of not having the universities in your country controlled by the government. When governments decide how to allocate resources, political deal-making causes things to be spread out geographically. No central goverment would put its two best universities in the same town, unless it was the capital (which would cause other problems). But scholars seem to like to cluster together as much as people in any other field, and when given the freedom to they derive the same advantages from it.

[2] There are still a few old professors in Palo Alto, but one by one they die and their houses are transformed by developers into McMansions and sold to VPs of Bus Dev.

[3] How many times have you read about startup founders who continued to live inexpensively as their companies took off? Who continued to dress in jeans and t-shirts, to drive the old car they had in grad school, and so on? If you did that in New York, people would treat you like shit. If you walk into a fancy restaurant in San Francisco wearing a jeans and a t-shirt, they're nice to you; who knows who you might be? Not in New York.

One sign of a city's potential as a technology center is the number of restaurants that still require jackets for men. According to Zagat's there are none in San Francisco, LA, Boston, or Seattle, 4 in DC, 6 in Chicago, 8 in London, 13 in New York, and 20 in Paris.

(Zagat's lists the Ritz Carlton Dining Room in SF as requiring jackets but I couldn't believe it, so I called to check and in fact they don't. Apparently there's only one restaurant left on the entire West Coast that still requires jackets: The French Laundry in Napa Valley.)

[4] Ideas are one step upstream from economic power, so it's conceivable that intellectual centers like Cambridge will one day have an edge over Silicon Valley like the one the Valley has over New York.

This seems unlikely at the moment; if anything Boston is falling further and further behind. The only reason I even mention the possibility is that the path from ideas to startups has recently been getting smoother. It's a lot easier now for a couple of hackers with no business experience to start a startup than it was 10 years ago. If you extrapolate another 20 years, maybe the balance of power will start to shift back. I wouldn't bet on it, but I wouldn't bet against it either.

[5] If Paris is where people care most about art, why is New York the center of gravity of the art business? Because in the twentieth century, art as brand split apart from art as stuff. New York is where the richest buyers are, but all they demand from art is brand, and since you can base brand on anything with a sufficiently identifiable style, you may as well use the local stuff.

Thanks to Trevor Blackwell, Sarah Harlin, Jessica Livingston, Jackie McDonough, Robert Morris, and David Sloo for reading drafts of this.

Italian Translation Portuguese Translation

Chinese Translation Korean Translation

097 The Pooled-Risk Company Management Company

July 2008

At this year's startup school, David Heinemeier Hansson gave a talk in which he

suggested that startup founders should do things the old fashioned way. Instead of hoping to get rich by building a valuable company and then selling stock in a "liquidity event," founders should start companies that make money and live off the revenues.

Sounds like a good plan. Let's think about the optimal way to do this.

One disadvantage of living off the revenues of your company is that you have to keep running it. And as anyone who runs their own business can tell you, that requires your complete attention. You can't just start a business and check out once things are going well, or they stop going well surprisingly fast.

The main economic motives of startup founders seem to be freedom and security. They want enough money that (a) they don't have to worry about running out of money and (b) they can spend their time how they want. Running your own business offers neither. You certainly don't have freedom: no boss is so demanding. Nor do you have security, because if you stop paying attention to the company, its revenues go away, and with them your income.

The best case, for most people, would be if you could hire someone to manage the company for you once you'd grown it to a certain size. Suppose you could find a really good manager. Then you would have both freedom and security. You could pay as little attention to the business as you wanted, knowing that your manager would keep things running smoothly. And that being so, revenues would continue to flow in, so you'd have security as well.

There will of course be some founders who wouldn't like that idea: the ones who like running their company so much that there's nothing else they'd rather do. But this group must be small. The way you succeed in most businesses is to be fanatically attentive to customers' needs. What are the odds that your own desires would coincide exactly with the demands of this powerful, external force?

Sure, running your own company can be fairly interesting. Viaweb was more interesting than any job I'd had before. And since I made much more money from it, it offered the highest ratio of income to boringness of anything I'd done, by orders of magnitude. But was it *the* most interesting work I could imagine doing? No.

Whether the number of founders in the same position is asymptotic or merely large, there are certainly a lot of them. For them the right approach would be to hand the company over to a professional manager eventually, if they could find one who was good enough.

So far so good. But what if your manager was hit by a bus? What you really want is a management company to run your company for you. Then you don't depend on any one person.

If you own rental property, there are companies you can hire to manage it for you. Some will do everything, from finding tenants to fixing leaks. Of course,

running companies is a lot more complicated than managing rental property, but let's suppose there were management companies that could do it for you. They'd charge a lot, but wouldn't it be worth it? I'd sacrifice a large percentage of the income for the extra peace of mind.

I realize what I'm describing already sounds too good to be true, but I can think of a way to make it even more attractive. If company management companies existed, there would be an additional service they could offer clients: they could let them insure their returns by pooling their risk. After all, even a perfect manager can't save a company when, as sometimes happens, its whole market dies, just as property managers can't save you from the building burning down. But a company that managed a large enough number of companies could say to all its clients: we'll combine the revenues from all your companies, and pay you your proportionate share.

If such management companies existed, they'd offer the maximum of freedom and security. Someone would run your company for you, and you'd be protected even if it happened to die.

Let's think about how such a management company might be organized. The simplest way would be to have a new kind of stock representing the total pool of companies they were managing. When you signed up, you'd trade your company's stock for shares of this pool, in proportion to an estimate of your company's value that you'd both agreed upon. Then you'd automatically get your share of the returns of the whole pool.

The catch is that because this kind of trade would be hard to undo, you couldn't switch management companies. But there's a way they could fix that: suppose all the company management companies got together and agreed to allow their clients to exchange shares in all their pools. Then you could, in effect, simultaneously choose all the management companies to run yours for you, in whatever proportion you wanted, and change your mind later as often as you wanted.

If such pooled-risk company management companies existed, signing up with one would seem the ideal plan for most people following the route David advocated.

Good news: they do exist. What I've just described is an acquisition by a public company.

Unfortunately, though public acquirers are structurally identical to pooled-risk company management companies, they don't think of themselves that way. With a property management company, you can just walk in whenever you want and say "manage my rental property for me" and they'll do it. Whereas acquirers are, as of this writing, extremely fickle. Sometimes they're in a buying mood and they'll overpay enormously; other times they're not interested. They're like property management companies run by madmen. Or more precisely, by Benjamin Graham's Mr. Market.

So while on average public acquirers behave like pooled-risk company managers, you need a window of several years to get average case performance. If you wait long enough (five years, say) you're likely to hit an up cycle where some acquirer is hot to buy you. But you can't choose when it happens.

You can't assume investors will carry you for as long as you might have to wait. Your company has to make money. Opinions are divided about how early to focus on that. Joe Kraus says you should try charging customers right away. And yet some of the most successful startups, including Google, ignored revenue at first and concentrated exclusively on development. The answer probably depends on the type of company you're starting. I can imagine some where trying to make sales would be a good heuristic for product design, and others where it would just be a distraction. The test is probably whether it helps you to understand your users.

You can choose whichever revenue strategy you think is best for the type of company you're starting, so long as you're profitable. Being profitable ensures you'll get at least the average of the acquisition market--in which public companies do behave as pooled-risk company management companies.

David isn't mistaken in saying you should start a company to live off its revenues. The mistake is thinking this is somehow opposed to starting a company and selling it. In fact, for most people the latter is merely the optimal case of the former.

Thanks to Trevor Blackwell, Jessica Livingston, Michael Mandel, Robert Morris, and Fred Wilson for reading drafts of this.

Russian Translation

100 The Other Half of "Artists Ship"

November 2008

One of the differences between big companies and startups is that big companies tend to have developed procedures to protect themselves against mistakes. A startup walks like a toddler, bashing into things and falling over all the time. A big company is more deliberate.

The gradual accumulation of checks in an organization is a kind of learning, based on disasters that have happened to it or others like it. After giving a contract to a supplier who goes bankrupt and fails to deliver, for example, a company might require all suppliers to prove they're solvent before submitting bids.

As companies grow they invariably get more such checks, either in response to disasters they've suffered, or (probably more often) by hiring people from bigger

companies who bring with them customs for protecting against new types of disasters.

It's natural for organizations to learn from mistakes. The problem is, people who propose new checks almost never consider that the check itself has a cost.

Every check has a cost. For example, consider the case of making suppliers verify their solvency. Surely that's mere prudence? But in fact it could have substantial costs. There's obviously the direct cost in time of the people on both sides who supply and check proofs of the supplier's solvency. But the real costs are the ones you never hear about: the company that would be the best supplier, but doesn't bid because they can't spare the effort to get verified. Or the company that would be the best supplier, but falls just short of the threshold for solvency--which will of course have been set on the high side, since there is no apparent cost of increasing it.

Whenever someone in an organization proposes to add a new check, they should have to explain not just the benefit but the cost. No matter how bad a job they did of analyzing it, this meta-check would at least remind everyone there had to be a cost, and send them looking for it.

If companies started doing that, they'd find some surprises. Joel Spolsky recently spoke at Y Combinator about selling software to corporate customers. He said that in most companies software costing up to about \$1000 could be bought by individual managers without any additional approvals. Above that threshold, software purchases generally had to be approved by a committee. But babysitting this process was so expensive for software vendors that it didn't make sense to charge less than \$50,000. Which means if you're making something you might otherwise have charged \$5000 for, you have to sell it for \$50,000 instead.

The purpose of the committee is presumably to ensure that the company doesn't waste money. And yet the result is that the company pays 10 times as much.

Checks on purchases will always be expensive, because the harder it is to sell something to you, the more it has to cost. And not merely linearly, either. If you're hard enough to sell to, the people who are best at making things don't want to bother. The only people who will sell to you are companies that specialize in selling to you. Then you've sunk to a whole new level of inefficiency. Market mechanisms no longer protect you, because the good suppliers are no longer in the market.

Such things happen constantly to the biggest organizations of all, governments. But checks instituted by governments can cause much worse problems than merely overpaying. Checks instituted by governments can cripple a country's whole economy. Up till about 1400, China was richer and more technologically advanced than Europe. One reason Europe pulled ahead was that the Chinese government restricted long trading voyages. So it was left to the Europeans to explore and eventually to dominate the rest of the world, including China.

In more recent times, Sarbanes-Oxley has practically destroyed the US IPO

market. That wasn't the intention of the legislators who wrote it. They just wanted to add a few more checks on public companies. But they forgot to consider the cost. They forgot that companies about to go public are usually rather stretched, and that the weight of a few extra checks that might be easy for General Electric to bear are enough to prevent younger companies from being public at all.

Once you start to think about the cost of checks, you can start to ask other interesting questions. Is the cost increasing or decreasing? Is it higher in some areas than others? Where does it increase discontinuously? If large organizations started to ask questions like that, they'd learn some frightening things.

I think the cost of checks may actually be increasing. The reason is that software plays an increasingly important role in companies, and the people who write software are particularly harmed by checks.

Programmers are unlike many types of workers in that the best ones actually prefer to work hard. This doesn't seem to be the case in most types of work. When I worked in fast food, we didn't prefer the busy times. And when I used to mow lawns, I definitely didn't prefer it when the grass was long after a week of rain.

Programmers, though, like it better when they write more code. Or more precisely, when they release more code. Programmers like to make a difference. Good ones, anyway.

For good programmers, one of the best things about working for a startup is that there are few checks on releases. In true startups, there are no external checks at all. If you have an idea for a new feature in the morning, you can write it and push it to the production servers before lunch. And when you can do that, you have more ideas.

At big companies, software has to go through various approvals before it can be launched. And the cost of doing this can be enormous--in fact, discontinuous. I was talking recently to a group of three programmers whose startup had been acquired a few years before by a big company. When they'd been independent, they could release changes instantly. Now, they said, the absolute fastest they could get code released on the production servers was two weeks.

This didn't merely make them less productive. It made them hate working for the acquirer.

Here's a sign of how much programmers like to be able to work hard: these guys would have *paid* to be able to release code immediately, the way they used to. I asked them if they'd trade 10% of the acquisition price for the ability to release code immediately, and all three instantly said yes. Then I asked what was the maximum percentage of the acquisition price they'd trade for it. They said they didn't want to think about it, because they didn't want to know how high they'd go, but I got the impression it might be as much as half.

They'd have sacrificed hundreds of thousands of dollars, perhaps millions, just to be able to deliver more software to users. And you know what? It would have been perfectly safe to let them. In fact, the acquirer would have been better off; not only wouldn't these guys have broken anything, they'd have gotten a lot more done. So the acquirer is in fact getting worse performance at greater cost. Just like the committee approving software purchases.

And just as the greatest danger of being hard to sell to is not that you overpay but that the best suppliers won't even sell to you, the greatest danger of applying too many checks to your programmers is not that you'll make them unproductive, but that good programmers won't even want to work for you.

Steve Jobs's famous maxim "artists ship" works both ways. Artists aren't merely capable of shipping. They insist on it. So if you don't let people ship, you won't have any artists.

105 Startups in 13 Sentences

Want to start a startup? Get funded by Y Combinator.

Watch how this essay was written.

February 2009

One of the things I always tell startups is a principle I learned from Paul Buchheit: it's better to make a few people really happy than to make a lot of people semi-happy. I was saying recently to a reporter that if I could only tell startups 10 things, this would be one of them. Then I thought: what would the other 9 be?

When I made the list there turned out to be 13:

1. Pick good cofounders.

Cofounders are for a startup what location is for real estate. You can change anything about a house except where it is. In a startup you can change your idea easily, but changing your cofounders is hard. 1 And the success of a startup is almost always a function of its founders.

2. Launch fast.

The reason to launch fast is not so much that it's critical to get your product to market early, but that you haven't really started working on it till you've launched. Launching teaches you what you should have been building. Till you know that you're wasting your time. So the main value of whatever you launch with is as a pretext for engaging users.

3. Let your idea evolve.

This is the second half of launching fast. Launch fast and iterate. It's a big mistake to treat a startup as if it were merely a matter of implementing some brilliant initial idea. As in an essay, most of the ideas appear in the implementing.

4. Understand your users.

You can envision the wealth created by a startup as a rectangle, where one side is the number of users and the other is how much you improve their lives. ² The second dimension is the one you have most control over. And indeed, the growth in the first will be driven by how well you do in the second. As in science, the hard part is not answering questions but asking them: the hard part is seeing something new that users lack. The better you understand them the better the odds of doing that. That's why so many successful startups make something the founders needed.

5. Better to make a few users love you than a lot ambivalent.

Ideally you want to make large numbers of users love you, but you can't expect to hit that right away. Initially you have to choose between satisfying all the needs of a subset of potential users, or satisfying a subset of the needs of all potential users. Take the first. It's easier to expand userwise than satisfactionwise. And perhaps more importantly, it's harder to lie to yourself. If you think you're 85% of the way to a great product, how do you know it's not 70%? Or 10%? Whereas it's easy to know how many users you have.

6. Offer surprisingly good customer service.

Customers are used to being maltreated. Most of the companies they deal with are quasi-monopolies that get away with atrocious customer service. Your own ideas about what's possible have been unconsciously lowered by such experiences. Try making your customer service not merely good, but surprisingly good. Go out of your way to make people happy. They'll be overwhelmed; you'll see. In the earliest stages of a startup, it pays to offer customer service on a level that wouldn't scale, because it's a way of learning about your users.

7. You make what you measure.

I learned this one from Joe Kraus. ³ Merely measuring something has an uncanny tendency to improve it. If you want to make your user numbers go up, put a big piece of paper on your wall and every day plot the number of users. You'll be delighted when it goes up and disappointed when it goes down. Pretty soon you'll start noticing what makes the number go up, and you'll start to do more of that. Corollary: be careful what you measure.

8. Spend little.

I can't emphasize enough how important it is for a startup to be cheap. Most startups fail before they make something people want, and the most common form of failure is running out of money. So being cheap is (almost) interchangeable with iterating rapidly. ⁴ But it's more than that. A culture of cheapness keeps companies young in something like the way exercise keeps people young.

9. Get ramen profitable.

"Ramen profitable" means a startup makes just enough to pay the founders' living expenses. It's not rapid prototyping for business models (though it can be), but more a way of hacking the investment process. Once you cross over into ramen profitable, it completely changes your relationship with investors. It's also great for morale.

10. Avoid distractions.

Nothing kills startups like distractions. The worst type are those that pay money: day jobs, consulting, profitable side-projects. The startup may have more long-term potential, but you'll always interrupt working on it to answer calls from people paying you now. Paradoxically, fundraising is this type of distraction, so try to minimize that too.

11. Don't get demoralized.

Though the immediate cause of death in a startup tends to be running out of money, the underlying cause is usually lack of focus. Either the company is run by stupid people (which can't be fixed with advice) or the people are smart but got demoralized. Starting a startup is a huge moral weight. Understand this and make a conscious effort not to be ground down by it, just as you'd be careful to bend at the knees when picking up a heavy box.

12. Don't give up.

Even if you get demoralized, don't give up. You can get surprisingly far by just not giving up. This isn't true in all fields. There are a lot of people who couldn't become good mathematicians no matter how long they persisted. But startups aren't like that. Sheer effort is usually enough, so long as you keep morphing your idea.

13. Deals fall through.

One of the most useful skills we learned from Viaweb was not getting our hopes up. We probably had 20 deals of various types fall through. After the first 10 or so we learned to treat deals as background processes that we should ignore till they terminated. It's very dangerous to morale to start to depend on deals closing, not just because they so often don't, but because it makes them less likely to.

Having gotten it down to 13 sentences, I asked myself which I'd choose if I could only keep one.

Understand your users. That's the key. The essential task in a startup is to create wealth; the dimension of wealth you have most control over is how much you improve users' lives; and the hardest part of that is knowing what to make for them. Once you know what to make, it's mere effort to make it, and most decent hackers are capable of that.

Understanding your users is part of half the principles in this list. That's the reason to launch early, to understand your users. Evolving your idea is the embodiment of understanding your users. Understanding your users well will tend to push you toward making something that makes a few people deeply happy. The most important reason for having surprisingly good customer service is that it helps you understand your users. And understanding your users will even ensure your morale, because when everything else is collapsing around you, having just ten users who love you will keep you going.

Notes

[1] Strictly speaking it's impossible without a time machine.

[2] In practice it's more like a ragged comb.

[3] Joe thinks one of the founders of Hewlett Packard said it first, but he doesn't remember which.

[4] They'd be interchangeable if markets stood still. Since they don't, working twice as fast is better than having twice as much time.

Turkish Translation Spanish Translation

Bulgarian Translation Japanese Translation

Persian Translation

110 Relentlessly Resourceful

Want to start a startup? Get funded by Y Combinator.

March 2009

A couple days ago I finally got being a good startup founder down to two words: relentlessly resourceful.

Till then the best I'd managed was to get the opposite quality down to one: hapless. Most dictionaries say hapless means unlucky. But the dictionaries are not doing a very good job. A team that outplays its opponents but loses because of a bad decision by the referee could be called unlucky, but not hapless. Hapless implies passivity. To be hapless is to be battered by circumstances--to let the world have its way with you, instead of having your way with the world. 1

Unfortunately there's no antonym of hapless, which makes it difficult to tell founders what to aim for. "Don't be hapless" is not much of rallying cry.

It's not hard to express the quality we're looking for in metaphors. The best is probably a running back. A good running back is not merely determined, but flexible as well. They want to get downfield, but they adapt their plans on the fly.

Unfortunately this is just a metaphor, and not a useful one to most people outside the US. "Be like a running back" is no better than "Don't be hapless."

But finally I've figured out how to express this quality directly. I was writing a talk for investors, and I had to explain what to look for in founders. What would someone who was the opposite of hapless be like? They'd be relentlessly resourceful. Not merely relentless. That's not enough to make things go your way except in a few mostly uninteresting domains. In any interesting domain, the difficulties will be novel. Which means you can't simply plow through them, because you don't know initially how hard they are; you don't know whether you're about to plow through a block of foam or granite. So you have to be resourceful. You have to keep trying new things.

Be relentlessly resourceful.

That sounds right, but is it simply a description of how to be successful in general? I don't think so. This isn't the recipe for success in writing or painting, for example. In that kind of work the recipe is more to be actively curious. Resourceful implies the obstacles are external, which they generally are in startups. But in writing and painting they're mostly internal; the obstacle is your own obtuseness. ²

There probably are other fields where "relentlessly resourceful" is the recipe for success. But though other fields may share it, I think this is the best short description we'll find of what makes a good startup founder. I doubt it could be made more precise.

Now that we know what we're looking for, that leads to other questions. For example, can this quality be taught? After four years of trying to teach it to people, I'd say that yes, surprisingly often it can. Not to everyone, but to many people. ³ Some people are just constitutionally passive, but others have a latent ability to be relentlessly resourceful that only needs to be brought out.

This is particularly true of young people who have till now always been under the thumb of some kind of authority. Being relentlessly resourceful is definitely not the recipe for success in big companies, or in most schools. I don't even want to think what the recipe is in big companies, but it is certainly longer and messier, involving some combination of resourcefulness, obedience, and building alliances.

Identifying this quality also brings us closer to answering a question people often wonder about: how many startups there could be. There is not, as some people seem to think, any economic upper bound on this number. There's no reason to believe there is any limit on the amount of newly created wealth consumers can absorb, any more than there is a limit on the number of theorems that can be proven. So probably the limiting factor on the number of startups is the pool of potential founders. Some people would make good founders, and others wouldn't. And now that we can say what makes a good founder, we know how to put an upper bound on the size of the pool.

This test is also useful to individuals. If you want to know whether you're the right sort of person to start a startup, ask yourself whether you're relentlessly resourceful. And if you want to know whether to recruit someone as a cofounder, ask if they are.

You can even use it tactically. If I were running a startup, this would be the phrase I'd tape to the mirror. "Make something people want" is the destination, but "Be relentlessly resourceful" is how you get there.

Notes

[1] I think the reason the dictionaries are wrong is that the meaning of the word has shifted. No one writing a dictionary from scratch today would say that hapless meant unlucky. But a couple hundred years ago they might have. People were more at the mercy of circumstances in the past, and as a result a lot of the words we use for good and bad outcomes have origins in words about luck.

When I was living in Italy, I was once trying to tell someone that I hadn't had much success in doing something, but I couldn't think of the Italian word for success. I spent some time trying to describe the word I meant. Finally she said "Ah! Fortuna!"

[2] There are aspects of startups where the recipe is to be actively curious. There can be times when what you're doing is almost pure discovery. Unfortunately these times are a small proportion of the whole. On the other hand, they are in research too.

[3] I'd almost say to most people, but I realize (a) I have no idea what most people are like, and (b) I'm pathologically optimistic about people's ability to change.

Thanks to Trevor Blackwell and Jessica Livingston for reading drafts of this.

115 Maker's Schedule, Manager's Schedule

"...the mere consciousness of an engagement will sometimes worry a whole day."

– Charles Dickens

July 2009

One reason programmers dislike meetings so much is that they're on a different type of schedule from other people. Meetings cost them more.

There are two types of schedule, which I'll call the manager's schedule and the maker's schedule. The manager's schedule is for bosses. It's embodied in the traditional appointment book, with each day cut into one hour intervals. You can block off several hours for a single task if you need to, but by default you change what you're doing every hour.

When you use time that way, it's merely a practical problem to meet with someone. Find an open slot in your schedule, book them, and you're done.

Most powerful people are on the manager's schedule. It's the schedule of command. But there's another way of using time that's common among people who make things, like programmers and writers. They generally prefer to use time in units of half a day at least. You can't write or program well in units of an hour. That's barely enough time to get started.

When you're operating on the maker's schedule, meetings are a disaster. A single meeting can blow a whole afternoon, by breaking it into two pieces each too small to do anything hard in. Plus you have to remember to go to the meeting. That's no problem for someone on the manager's schedule. There's always something coming on the next hour; the only question is what. But when someone on the maker's schedule has a meeting, they have to think about it.

For someone on the maker's schedule, having a meeting is like throwing an exception. It doesn't merely cause you to switch from one task to another; it changes the mode in which you work.

I find one meeting can sometimes affect a whole day. A meeting commonly blows at least half a day, by breaking up a morning or afternoon. But in addition there's sometimes a cascading effect. If I know the afternoon is going to be broken up, I'm slightly less likely to start something ambitious in the morning. I know this may sound oversensitive, but if you're a maker, think of your own case. Don't your spirits rise at the thought of having an entire day free to work, with no appointments at all? Well, that means your spirits are correspondingly depressed when you don't. And ambitious projects are by definition close to the limits of your capacity. A small decrease in morale is enough to kill them off.

Each type of schedule works fine by itself. Problems arise when they meet. Since most powerful people operate on the manager's schedule, they're in a position to make everyone resonate at their frequency if they want to. But the smarter ones restrain themselves, if they know that some of the people working for them need long chunks of time to work in.

Our case is an unusual one. Nearly all investors, including all VCs I know, operate on the manager's schedule. But Y Combinator runs on the maker's schedule. Rtm and Trevor and I do because we always have, and Jessica does too, mostly, because she's gotten into sync with us.

I wouldn't be surprised if there start to be more companies like us. I suspect founders may increasingly be able to resist, or at least postpone, turning into managers, just as a few decades ago they started to be able to resist switching from jeans to suits.

How do we manage to advise so many startups on the maker's schedule? By using the classic device for simulating the manager's schedule within the maker's: office hours. Several times a week I set aside a chunk of time to meet founders we've funded. These chunks of time are at the end of my working day, and I

wrote a signup program that ensures all the appointments within a given set of office hours are clustered at the end. Because they come at the end of my day these meetings are never an interruption. (Unless their working day ends at the same time as mine, the meeting presumably interrupts theirs, but since they made the appointment it must be worth it to them.) During busy periods, office hours sometimes get long enough that they compress the day, but they never interrupt it.

When we were working on our own startup, back in the 90s, I evolved another trick for partitioning the day. I used to program from dinner till about 3 am every day, because at night no one could interrupt me. Then I'd sleep till about 11 am, and come in and work until dinner on what I called "business stuff." I never thought of it in these terms, but in effect I had two workdays each day, one on the manager's schedule and one on the maker's.

When you're operating on the manager's schedule you can do something you'd never want to do on the maker's: you can have speculative meetings. You can meet someone just to get to know one another. If you have an empty slot in your schedule, why not? Maybe it will turn out you can help one another in some way.

Business people in Silicon Valley (and the whole world, for that matter) have speculative meetings all the time. They're effectively free if you're on the manager's schedule. They're so common that there's distinctive language for proposing them: saying that you want to "grab coffee," for example.

Speculative meetings are terribly costly if you're on the maker's schedule, though. Which puts us in something of a bind. Everyone assumes that, like other investors, we run on the manager's schedule. So they introduce us to someone they think we ought to meet, or send us an email proposing we grab coffee. At this point we have two options, neither of them good: we can meet with them, and lose half a day's work; or we can try to avoid meeting them, and probably offend them.

Till recently we weren't clear in our own minds about the source of the problem. We just took it for granted that we had to either blow our schedules or offend people. But now that I've realized what's going on, perhaps there's a third option: to write something explaining the two types of schedule. Maybe eventually, if the conflict between the manager's schedule and the maker's schedule starts to be more widely understood, it will become less of a problem.

Those of us on the maker's schedule are willing to compromise. We know we have to have some number of meetings. All we ask from those on the manager's schedule is that they understand the cost.

Thanks to Sam Altman, Trevor Blackwell, Paul Buchheit, Jessica Livingston, and Robert Morris for reading drafts of this.

Related:

How to Do What You Love Good and Bad Procrastination

116 Ramen Profitable

Want to start a startup? Get funded by Y Combinator.

July 2009

Now that the term "ramen profitable" has become widespread, I ought to explain precisely what the idea entails.

Ramen profitable means a startup makes just enough to pay the founders' living expenses. This is a different form of profitability than startups have traditionally aimed for. Traditional profitability means a big bet is finally paying off, whereas the main importance of ramen profitability is that it buys you time. ¹

In the past, a startup would usually become profitable only after raising and spending quite a lot of money. A company making computer hardware might not become profitable for 5 years, during which they spent \$50 million. But when they did they might have revenues of \$50 million a year. This kind of profitability means the startup has succeeded.

Ramen profitability is the other extreme: a startup that becomes profitable after 2 months, even though its revenues are only \$3000 a month, because the only employees are a couple 25 year old founders who can live on practically nothing. Revenues of \$3000 a month do not mean the company has succeeded. But it does share something with the one that's profitable in the traditional way: they don't need to raise money to survive.

Ramen profitability is an unfamiliar idea to most people because it only recently became feasible. It's still not feasible for a lot of startups; it would not be for most biotech startups, for example; but it is for many software startups because they're now so cheap. For many, the only real cost is the founders' living expenses.

The main significance of this type of profitability is that you're no longer at the mercy of investors. If you're still losing money, then eventually you'll either have to raise more or shut down. Once you're ramen profitable this painful choice goes away. You can still raise money, but you don't have to do it now.

The most obvious advantage of not needing money is that you can get better terms. If investors know you need money, they'll sometimes take advantage of you. Some may even deliberately stall, because they know that as you run out of money you'll become increasingly pliable.

But there are also three less obvious advantages of ramen profitability. One is that it makes you more attractive to investors. If you're already profitable, on however small a scale, it shows that (a) you can get at least someone to pay you, (b) you're serious about building things people want, and (c) you're disciplined enough to keep expenses low.

This is reassuring to investors, because you've addressed three of their biggest worries. It's common for them to fund companies that have smart founders and a big market, and yet still fail. When these companies fail, it's usually because (a) people wouldn't pay for what they made, e.g. because it was too hard to sell to them, or the market wasn't ready yet, (b) the founders solved the wrong problem, instead of paying attention to what users needed, or (c) the company spent too much and burned through their funding before they started to make money. If you're ramen profitable, you're already avoiding these mistakes.

Another advantage of ramen profitability is that it's good for morale. A company tends to feel rather theoretical when you first start it. It's legally a company, but you feel like you're lying when you call it one. When people start to pay you significant amounts, the company starts to feel real. And your own living expenses are the milestone you feel most, because at that point the future flips state. Now survival is the default, instead of dying.

A morale boost on that scale is very valuable in a startup, because the moral weight of running a startup is what makes it hard. Startups are still very rare. Why don't more people do it? The financial risk? Plenty of 25 year olds save nothing anyway. The long hours? Plenty of people work just as long hours in regular jobs. What keeps people from starting startups is the fear of having so much responsibility. And this is not an irrational fear: it really is hard to bear. Anything that takes some of that weight off you will greatly increase your chances of surviving.

A startup that reaches ramen profitability may be more likely to succeed than not. Which is pretty exciting, considering the bimodal distribution of outcomes in startups: you either fail or make a lot of money.

The fourth advantage of ramen profitability is the least obvious but may be the most important. If you don't need to raise money, you don't have to interrupt working on the company to do it.

Raising money is terribly distracting. You're lucky if your productivity is a third of what it was before. And it can last for months.

I didn't understand (or rather, remember) precisely why raising money was so distracting till earlier this year. I'd noticed that startups we funded would usually grind to a halt when they switched to raising money, but I didn't remember exactly why till YC raised money itself. We had a comparatively easy time of it; the first people I asked said yes; but it took months to work out the details, and during that time I got hardly any real work done. Why? Because I thought about it all the time.

At any given time there tends to be one problem that's the most urgent for a startup. This is what you think about as you fall asleep at night and when you take a shower in the morning. And when you start raising money, that becomes the problem you think about. You only take one shower in the morning, and if you're thinking about investors during it, then you're not thinking about the product.

Whereas if you can choose when you raise money, you can pick a time when you're not in the middle of something else, and you can probably also insist that the round close fast. You may even be able to avoid having the round occupy your thoughts, if you don't care whether it closes.

Ramen profitable means no more than the definition implies. It does not, for example, imply that you're "bootstrapping" the startup--that you're never going to take money from investors. Empirically that doesn't seem to work very well. Few startups succeed without taking investment. Maybe as startups get cheaper it will become more common. On the other hand, the money is there, waiting to be invested. If startups need it less, they'll be able to get it on better terms, which will make them more inclined to take it. That will tend to produce an equilibrium. ²

Another thing ramen profitability doesn't imply is Joe Kraus's idea that you should put your business model in beta when you put your product in beta. He believes you should get people to pay you from the beginning. I think that's too constraining. Facebook didn't, and they've done better than most startups. Making money right away was not only unnecessary for them, but probably would have been harmful. I do think Joe's rule could be useful for many startups, though. When founders seem unfocused, I sometimes suggest they try to get customers to pay them for something, in the hope that this constraint will prod them into action.

The difference between Joe's idea and ramen profitability is that a ramen profitable company doesn't have to be making money the way it ultimately will. It just has to be making money. The most famous example is Google, which initially made money by licensing search to sites like Yahoo.

Is there a downside to ramen profitability? Probably the biggest danger is that it might turn you into a consulting firm. Startups have to be product companies, in the sense of making a single thing that everyone uses. The defining quality of startups is that they grow fast, and consulting just can't scale the way a product can. ³ But it's pretty easy to make \$3000 a month consulting; in fact, that would be a low rate for contract programming. So there could be a temptation to slide into consulting, and telling yourselves you're a ramen profitable startup, when in fact you're not a startup at all.

It's ok to do a little consulting-type work at first. Startups usually have to do something weird at first. But remember that ramen profitability is not the

destination. A startup's destination is to grow really big; ramen profitability is a trick for not dying en route.

Notes

[1] The "ramen" in "ramen profitable" refers to instant ramen, which is just about the cheapest food available.

Please do not take the term literally. Living on instant ramen would be very unhealthy. Rice and beans are a better source of food. Start by investing in a rice cooker, if you don't have one.

Rice and Beans for 2n

[code]

```
olive oil or butter      n yellow onions      other fresh vegetables; experiment
```

[/code]

Put rice in rice cooker. Add water as specified on rice package. (Default: 2 cups water per cup of rice.) Turn on rice cooker and forget about it.

Chop onions and other vegetables and fry in oil, over fairly low heat, till onions are glassy. Put in chopped garlic, pepper, cumin, and a little more fat, and stir. Keep heat low. Cook another 2 or 3 minutes, then add beans (don't drain the beans), and stir. Throw in the bouillon cube(s), cover, and cook on lowish heat for at least 10 minutes more. Stir vigilantly to avoid sticking.

If you want to save money, buy beans in giant cans from discount stores. Spices are also much cheaper when bought in bulk. If there's an Indian grocery store near you, they'll have big bags of cumin for the same price as the little jars in supermarkets.

[2] There's a good chance that a shift in power from investors to founders would actually increase the size of the venture business. I think investors currently err too far on the side of being harsh to founders. If they were forced to stop, the whole venture business would work better, and you might see something like the increase in trade you always see when restrictive laws are removed.

Investors are one of the biggest sources of pain for founders; if they stopped causing so much pain, it would be better to be a founder; and if it were better to be a founder, more people would do it.

[3] It's conceivable that a startup could grow big by transforming consulting into a form that would scale. But if they did that they'd really be a product company.

Thanks to Jessica Livingston for reading drafts of this.

Japanese Translation

119 The Anatomy of Determination

Want to start a startup? Get funded by Y Combinator.

September 2009

Like all investors, we spend a lot of time trying to learn how to predict which startups will succeed. We probably spend more time thinking about it than most, because we invest the earliest. Prediction is usually all we have to rely on.

We learned quickly that the most important predictor of success is determination. At first we thought it might be intelligence. Everyone likes to believe that's what makes startups succeed. It makes a better story that a company won because its founders were so smart. The PR people and reporters who spread such stories probably believe them themselves. But while it certainly helps to be smart, it's not the deciding factor. There are plenty of people as smart as Bill Gates who achieve nothing.

In most domains, talent is overrated compared to determination--partly because it makes a better story, partly because it gives onlookers an excuse for being lazy, and partly because after a while determination starts to look like talent.

I can't think of any field in which determination is overrated, but the relative importance of determination and talent probably do vary somewhat. Talent probably matters more in types of work that are purer, in the sense that one is solving mostly a single type of problem instead of many different types. I suspect determination would not take you as far in math as it would in, say, organized crime.

I don't mean to suggest by this comparison that types of work that depend more on talent are always more admirable. Most people would agree it's more admirable to be good at math than memorizing long strings of digits, even though the latter depends more on natural ability.

Perhaps one reason people believe startup founders win by being smarter is that intelligence does matter more in technology startups than it used to in earlier types of companies. You probably do need to be a bit smarter to dominate Internet search than you had to be to dominate railroads or hotels or newspapers. And that's probably an ongoing trend. But even in the highest of high tech industries, success still depends more on determination than brains.

If determination is so important, can we isolate its components? Are some more important than others? Are there some you can cultivate?

The simplest form of determination is sheer willfulness. When you want something, you must have it, no matter what.

A good deal of willfulness must be inborn, because it's common to see families where one sibling has much more of it than another. Circumstances can alter it, but at the high end of the scale, nature seems to be more important than nurture. Bad circumstances can break the spirit of a strong-willed person, but

I don't think there's much you can do to make a weak-willed person stronger-willed.

Being strong-willed is not enough, however. You also have to be hard on yourself. Someone who was strong-willed but self-indulgent would not be called determined. Determination implies your willfulness is balanced by discipline.

That word balance is a significant one. The more willful you are, the more disciplined you have to be. The stronger your will, the less anyone will be able to argue with you except yourself. And someone has to argue with you, because everyone has base impulses, and if you have more will than discipline you'll just give into them and end up on a local maximum like drug addiction.

We can imagine will and discipline as two fingers squeezing a slippery melon seed. The harder they squeeze, the further the seed flies, but they must both squeeze equally or the seed spins off sideways.

If this is true it has interesting implications, because discipline can be cultivated, and in fact does tend to vary quite a lot in the course of an individual's life. If determination is effectively the product of will and discipline, then you can become more determined by being more disciplined. 1

Another consequence of the melon seed model is that the more willful you are, the more dangerous it is to be undisciplined. There seem to be plenty of examples to confirm that. In some very energetic people's lives you see something like wing flutter, where they alternate between doing great work and doing absolutely nothing. Externally this would look a lot like bipolar disorder.

The melon seed model is inaccurate in at least one respect, however: it's static. In fact the dangers of indiscipline increase with temptation. Which means, interestingly, that determination tends to erode itself. If you're sufficiently determined to achieve great things, this will probably increase the number of temptations around you. Unless you become proportionally more disciplined, willfulness will then get the upper hand, and your achievement will revert to the mean.

That's why Shakespeare's Caesar thought thin men so dangerous. They weren't tempted by the minor perquisites of power.

The melon seed model implies it's possible to be too disciplined. Is it? I think there probably are people whose willfulness is crushed down by excessive discipline, and who would achieve more if they weren't so hard on themselves. One reason the young sometimes succeed where the old fail is that they don't realize how incompetent they are. This lets them do a kind of deficit spending. When they first start working on something, they overrate their achievements. But that gives them confidence to keep working, and their performance improves. Whereas someone clearer-eyed would see their initial incompetence for what it was, and perhaps be discouraged from continuing.

There's one other major component of determination: ambition. If willfulness

and discipline are what get you to your destination, ambition is how you choose it.

I don't know if it's exactly right to say that ambition is a component of determination, but they're not entirely orthogonal. It would seem a misnomer if someone said they were very determined to do something trivially easy.

And fortunately ambition seems to be quite malleable; there's a lot you can do to increase it. Most people don't know how ambitious to be, especially when they're young. They don't know what's hard, or what they're capable of. And this problem is exacerbated by having few peers. Ambitious people are rare, so if everyone is mixed together randomly, as they tend to be early in people's lives, then the ambitious ones won't have many ambitious peers. When you take people like this and put them together with other ambitious people, they bloom like dying plants given water. Probably most ambitious people are starved for the sort of encouragement they'd get from ambitious peers, whatever their age.

2

Achievements also tend to increase your ambition. With each step you gain confidence to stretch further next time.

So here in sum is how determination seems to work: it consists of willfulness balanced with discipline, aimed by ambition. And fortunately at least two of these three qualities can be cultivated. You may be able to increase your strength of will somewhat; you can definitely learn self-discipline; and almost everyone is practically malnourished when it comes to ambition.

I feel like I understand determination a bit better now. But only a bit: willfulness, discipline, and ambition are all concepts almost as complicated as determination.

3

Note too that determination and talent are not the whole story. There's a third factor in achievement: how much you like the work. If you really love working on something, you don't need determination to drive you; it's what you'd do anyway. But most types of work have aspects one doesn't like, because most types of work consist of doing things for other people, and it's very unlikely that the tasks imposed by their needs will happen to align exactly with what you want to do.

Indeed, if you want to create the most wealth, the way to do it is to focus more on their needs than your interests, and make up the difference with determination.

Notes

[1] Loosely speaking. What I'm claiming with the melon seed model is more like determination is proportionate to $w^m - k|w - d|^n$, where w is will and d discipline.

[2] Which means one of the best ways to help a society generally is to create events and institutions that bring ambitious people together. It's like pulling the control

rods out of a reactor: the energy they emit encourages other ambitious people, instead of being absorbed by the normal people they're usually surrounded with.

Conversely, it's probably a mistake to do as some European countries have done and try to ensure none of your universities is significantly better than the others.

[3] For example, willfulness clearly has two subcomponents, stubbornness and energy. The first alone yields someone who's stubbornly inert. The second alone yields someone flighty. As willful people get older or otherwise lose their energy, they tend to become merely stubborn.

Thanks to Sam Altman, Jessica Livingston, and Robert Morris for reading drafts of this.

Italian Translation Portuguese Translation

Russian Translation

125 Organic Startup Ideas

Want to start a startup? Get funded by Y Combinator.

April 2010

The best way to come up with startup ideas is to ask yourself the question: what do you wish someone would make for you?

There are two types of startup ideas: those that grow organically out of your own life, and those that you decide, from afar, are going to be necessary to some class of users other than you. Apple was the first type. Apple happened because Steve Wozniak wanted a computer. Unlike most people who wanted computers, he could design one, so he did. And since lots of other people wanted the same thing, Apple was able to sell enough of them to get the company rolling. They still rely on this principle today, incidentally. The iPhone is the phone Steve Jobs wants. 1

Our own startup, Viaweb, was of the second type. We made software for building online stores. We didn't need this software ourselves. We weren't direct marketers. We didn't even know when we started that our users were called "direct marketers." But we were comparatively old when we started the company (I was 30 and Robert Morris was 29), so we'd seen enough to know users would need this type of software. 2

There is no sharp line between the two types of ideas, but the most successful startups seem to be closer to the Apple type than the Viaweb type. When he was writing that first Basic interpreter for the Altair, Bill Gates was writing something he would use, as were Larry and Sergey when they wrote the first versions of Google.

Organic ideas are generally preferable to the made up kind, but particularly so when the founders are young. It takes experience to predict what other people will want. The worst ideas we see at Y Combinator are from young founders making things they think other people will want.

So if you want to start a startup and don't know yet what you're going to do, I'd encourage you to focus initially on organic ideas. What's missing or broken in your daily life? Sometimes if you just ask that question you'll get immediate answers. It must have seemed obviously broken to Bill Gates that you could only program the Altair in machine language.

You may need to stand outside yourself a bit to see brokenness, because you tend to get used to it and take it for granted. You can be sure it's there, though. There are always great ideas sitting right under our noses. In 2004 it was ridiculous that Harvard undergrads were still using a Facebook printed on paper. Surely that sort of thing should have been online.

There are ideas that obvious lying around now. The reason you're overlooking them is the same reason you'd have overlooked the idea of building Facebook in 2004: organic startup ideas usually don't seem like startup ideas at first. We know now that Facebook was very successful, but put yourself back in 2004. Putting undergraduates' profiles online wouldn't have seemed like much of a startup idea. And in fact, it wasn't initially a startup idea. When Mark spoke at a YC dinner this winter he said he wasn't trying to start a company when he wrote the first version of Facebook. It was just a project. So was the Apple I when Woz first started working on it. He didn't think he was starting a company. If these guys had thought they were starting companies, they might have been tempted to do something more "serious," and that would have been a mistake.

So if you want to come up with organic startup ideas, I'd encourage you to focus more on the idea part and less on the startup part. Just fix things that seem broken, regardless of whether it seems like the problem is important enough to build a company on. If you keep pursuing such threads it would be hard not to end up making something of value to a lot of people, and when you do, surprise, you've got a company. 3

Don't be discouraged if what you produce initially is something other people dismiss as a toy. In fact, that's a good sign. That's probably why everyone else has been overlooking the idea. The first microcomputers were dismissed as toys. And the first planes, and the first cars. At this point, when someone comes to us with something that users like but that we could envision forum trolls dismissing as a toy, it makes us especially likely to invest.

While young founders are at a disadvantage when coming up with made-up ideas, they're the best source of organic ones, because they're at the forefront of technology. They use the latest stuff. They only just decided what to use, so why wouldn't they? And because they use the latest stuff, they're in a position to discover valuable types of fixable brokenness first.

There's nothing more valuable than an unmet need that is just becoming fixable. If you find something broken that you can fix for a lot of people, you've found a gold mine. As with an actual gold mine, you still have to work hard to get the gold out of it. But at least you know where the seam is, and that's the hard part.

Notes

[1] This suggests a way to predict areas where Apple will be weak: things Steve Jobs doesn't use. E.g. I doubt he is much into gaming.

[2] In retrospect, we should have *become* direct marketers. If I were doing Viaweb again, I'd open our own online store. If we had, we'd have understood users a lot better. I'd encourage anyone starting a startup to become one of its users, however unnatural it seems.

[3] Possible exception: It's hard to compete directly with open source software. You can build things for programmers, but there has to be some part you can charge for.

Thanks to Sam Altman, Trevor Blackwell, and Jessica Livingston for reading drafts of this.

126 How to Lose Time and Money

July 2010

When we sold our startup in 1998 I suddenly got a lot of money. I now had to think about something I hadn't had to think about before: how not to lose it. I knew it was possible to go from rich to poor, just as it was possible to go from poor to rich. But while I'd spent a lot of the past several years studying the paths from poor to rich, I knew practically nothing about the paths from rich to poor. Now, in order to avoid them, I had to learn where they were.

So I started to pay attention to how fortunes are lost. If you'd asked me as a kid how rich people became poor, I'd have said by spending all their money. That's how it happens in books and movies, because that's the colorful way to do it. But in fact the way most fortunes are lost is not through excessive expenditure, but through bad investments.

It's hard to spend a fortune without noticing. Someone with ordinary tastes would find it hard to blow through more than a few tens of thousands of dollars without thinking "wow, I'm spending a lot of money." Whereas if you start trading derivatives, you can lose a million dollars (as much as you want, really) in the blink of an eye.

In most people's minds, spending money on luxuries sets off alarms that making investments doesn't. Luxuries seem self-indulgent. And unless you got the money

by inheriting it or winning a lottery, you've already been thoroughly trained that self-indulgence leads to trouble. Investing bypasses those alarms. You're not spending the money; you're just moving it from one asset to another. Which is why people trying to sell you expensive things say "it's an investment."

The solution is to develop new alarms. This can be a tricky business, because while the alarms that prevent you from overspending are so basic that they may even be in our DNA, the ones that prevent you from making bad investments have to be learned, and are sometimes fairly counterintuitive.

A few days ago I realized something surprising: the situation with time is much the same as with money. The most dangerous way to lose time is not to spend it having fun, but to spend it doing fake work. When you spend time having fun, you know you're being self-indulgent. Alarms start to go off fairly quickly. If I woke up one morning and sat down on the sofa and watched TV all day, I'd feel like something was terribly wrong. Just thinking about it makes me wince. I'd start to feel uncomfortable after sitting on a sofa watching TV for 2 hours, let alone a whole day.

And yet I've definitely had days when I might as well have sat in front of a TV all day -- days at the end of which, if I asked myself what I got done that day, the answer would have been: basically, nothing. I feel bad after these days too, but nothing like as bad as I'd feel if I spent the whole day on the sofa watching TV. If I spent a whole day watching TV I'd feel like I was descending into perdition. But the same alarms don't go off on the days when I get nothing done, because I'm doing stuff that seems, superficially, like real work. Dealing with email, for example. You do it sitting at a desk. It's not fun. So it must be work.

With time, as with money, avoiding pleasure is no longer enough to protect you. It probably was enough to protect hunter-gatherers, and perhaps all pre-industrial societies. So nature and nurture combine to make us avoid self-indulgence. But the world has gotten more complicated: the most dangerous traps now are new behaviors that bypass our alarms about self-indulgence by mimicking more virtuous types. And the worst thing is, they're not even fun.

Thanks to Sam Altman, Trevor Blackwell, Patrick Collison, Jessica Livingston, and Robert Morris for reading drafts of this.

127 The Top Idea in Your Mind

Want to start a startup? Get funded by Y Combinator.

July 2010

I realized recently that what one thinks about in the shower in the morning is more important than I'd thought. I knew it was a good time to have ideas. Now

I'd go further: now I'd say it's hard to do a really good job on anything you don't think about in the shower.

Everyone who's worked on difficult problems is probably familiar with the phenomenon of working hard to figure something out, failing, and then suddenly seeing the answer a bit later while doing something else. There's a kind of thinking you do without trying to. I'm increasingly convinced this type of thinking is not merely helpful in solving hard problems, but necessary. The tricky part is, you can only control it indirectly. 1

I think most people have one top idea in their mind at any given time. That's the idea their thoughts will drift toward when they're allowed to drift freely. And this idea will thus tend to get all the benefit of that type of thinking, while others are starved of it. Which means it's a disaster to let the wrong idea become the top one in your mind.

What made this clear to me was having an idea I didn't want as the top one in my mind for two long stretches.

I'd noticed startups got way less done when they started raising money, but it was not till we ourselves raised money that I understood why. The problem is not the actual time it takes to meet with investors. The problem is that once you start raising money, raising money becomes the top idea in your mind. That becomes what you think about when you take a shower in the morning. And that means other questions aren't.

I'd hated raising money when I was running Viaweb, but I'd forgotten why I hated it so much. When we raised money for Y Combinator, I remembered. Money matters are particularly likely to become the top idea in your mind. The reason is that they have to be. It's hard to get money. It's not the sort of thing that happens by default. It's not going to happen unless you let it become the thing you think about in the shower. And then you'll make little progress on anything else you'd rather be working on. 2

(I hear similar complaints from friends who are professors. Professors nowadays seem to have become professional fundraisers who do a little research on the side. It may be time to fix that.)

The reason this struck me so forcibly is that for most of the preceding 10 years I'd been able to think about what I wanted. So the contrast when I couldn't was sharp. But I don't think this problem is unique to me, because just about every startup I've seen grinds to a halt when they start raising money — or talking to acquirers.

You can't directly control where your thoughts drift. If you're controlling them, they're not drifting. But you can control them indirectly, by controlling what situations you let yourself get into. That has been the lesson for me: be careful what you let become critical to you. Try to get yourself into situations where the most urgent problems are ones you want to think about.

You don't have complete control, of course. An emergency could push other thoughts out of your head. But barring emergencies you have a good deal of indirect control over what becomes the top idea in your mind.

I've found there are two types of thoughts especially worth avoiding — thoughts like the Nile Perch in the way they push out more interesting ideas. One I've already mentioned: thoughts about money. Getting money is almost by definition an attention sink. The other is disputes. These too are engaging in the wrong way: they have the same velcro-like shape as genuinely interesting ideas, but without the substance. So avoid disputes if you want to get real work done. 3

Even Newton fell into this trap. After publishing his theory of colors in 1672 he found himself distracted by disputes for years, finally concluding that the only solution was to stop publishing:

I see I have made myself a slave to Philosophy, but if I get free of Mr > Linus's business I will resolutely bid adieu to it eternally, excepting what > I do for my private satisfaction or leave to come out after me. For I see a > man must either resolve to put out nothing new or become a slave to defend > it. 4

Linus and his students at Liege were among the more tenacious critics. Newton's biographer Westfall seems to feel he was overreacting:

Recall that at the time he wrote, Newton's "slavery" consisted of five > replies to Liege, totalling fourteen printed pages, over the course of a > year.

I'm more sympathetic to Newton. The problem was not the 14 pages, but the pain of having this stupid controversy constantly reintroduced as the top idea in a mind that wanted so eagerly to think about other things.

Turning the other cheek turns out to have selfish advantages. Someone who does you an injury hurts you twice: first by the injury itself, and second by taking up your time afterward thinking about it. If you learn to ignore injuries you can at least avoid the second half. I've found I can to some extent avoid thinking about nasty things people have done to me by telling myself: this doesn't deserve space in my head. I'm always delighted to find I've forgotten the details of disputes, because that means I hadn't been thinking about them. My wife thinks I'm more forgiving than she is, but my motives are purely selfish.

I suspect a lot of people aren't sure what's the top idea in their mind at any given time. I'm often mistaken about it. I tend to think it's the idea I'd want to be the top one, rather than the one that is. But it's easy to figure this out: just take a shower. What topic do your thoughts keep returning to? If it's not what you want to be thinking about, you may want to change something.

Notes

[1] No doubt there are already names for this type of thinking, but I call it "ambient thought."

[2] This was made particularly clear in our case, because neither of the funds we raised was difficult, and yet in both cases the process dragged on for months. Moving large amounts of money around is never something people treat casually. The attention required increases with the amount--maybe not linearly, but definitely monotonically.

[3] Corollary: Avoid becoming an administrator, or your job will consist of dealing with money and disputes.

[4] Letter to Oldenburg, quoted in Westfall, Richard, *Life of Isaac Newton*, p. 107.

Thanks to Sam Altman, Patrick Collison, Jessica Livingston, and Robert Morris for reading drafts of this.

128 The Acceleration of Addictiveness

July 2010

What hard liquor, cigarettes, heroin, and crack have in common is that they're all more concentrated forms of less addictive predecessors. Most if not all the things we describe as addictive are. And the scary thing is, the process that created them is accelerating.

We wouldn't want to stop it. It's the same process that cures diseases: technological progress. Technological progress means making things do more of what we want. When the thing we want is something we want to want, we consider technological progress good. If some new technique makes solar cells $x\%$ more efficient, that seems strictly better. When progress concentrates something we don't want to want -- when it transforms opium into heroin -- it seems bad. But it's the same process at work. ¹

No one doubts this process is accelerating, which means increasing numbers of things we like will be transformed into things we like too much. ²

As far as I know there's no word for something we like too much. The closest is the colloquial sense of "addictive." That usage has become increasingly common during my lifetime. And it's clear why: there are an increasing number of things we need it for. At the extreme end of the spectrum are crack and meth. Food has been transformed by a combination of factory farming and innovations in food processing into something with way more immediate bang for the buck, and you can see the results in any town in America. Checkers and solitaire have been replaced by World of Warcraft and FarmVille. TV has become much more engaging, and even so it can't compete with Facebook.

The world is more addictive than it was 40 years ago. And unless the forms of technological progress that produced these things are subject to different laws

than technological progress in general, the world will get more addictive in the next 40 years than it did in the last 40.

The next 40 years will bring us some wonderful things. I don't mean to imply they're all to be avoided. Alcohol is a dangerous drug, but I'd rather live in a world with wine than one without. Most people can coexist with alcohol; but you have to be careful. More things we like will mean more things we have to be careful about.

Most people won't, unfortunately. Which means that as the world becomes more addictive, the two senses in which one can live a normal life will be driven ever further apart. One sense of "normal" is statistically normal: what everyone else does. The other is the sense we mean when we talk about the normal operating range of a piece of machinery: what works best.

These two senses are already quite far apart. Already someone trying to live well would seem eccentrically abstemious in most of the US. That phenomenon is only going to become more pronounced. You can probably take it as a rule of thumb from now on that if people don't think you're weird, you're living badly.

Societies eventually develop antibodies to addictive new things. I've seen that happen with cigarettes. When cigarettes first appeared, they spread the way an infectious disease spreads through a previously isolated population. Smoking rapidly became a (statistically) normal thing. There were ashtrays everywhere. We had ashtrays in our house when I was a kid, even though neither of my parents smoked. You had to for guests.

As knowledge spread about the dangers of smoking, customs changed. In the last 20 years, smoking has been transformed from something that seemed totally normal into a rather seedy habit: from something movie stars did in publicity shots to something small huddles of addicts do outside the doors of office buildings. A lot of the change was due to legislation, of course, but the legislation couldn't have happened if customs hadn't already changed.

It took a while though--on the order of 100 years. And unless the rate at which social antibodies evolve can increase to match the accelerating rate at which technological progress throws off new addictions, we'll be increasingly unable to rely on customs to protect us. ³ Unless we want to be canaries in the coal mine of each new addiction--the people whose sad example becomes a lesson to future generations--we'll have to figure out for ourselves what to avoid and how. It will actually become a reasonable strategy (or a more reasonable strategy) to suspect everything new.

In fact, even that won't be enough. We'll have to worry not just about new things, but also about existing things becoming more addictive. That's what bit me. I've avoided most addictions, but the Internet got me because it became addictive while I was using it. ⁴

Most people I know have problems with Internet addiction. We're all trying to figure out our own customs for getting free of it. That's why I don't have an

iPhone, for example; the last thing I want is for the Internet to follow me out into the world. 5 My latest trick is taking long hikes. I used to think running was a better form of exercise than hiking because it took less time. Now the slowness of hiking seems an advantage, because the longer I spend on the trail, the longer I have to think without interruption.

Sounds pretty eccentric, doesn't it? It always will when you're trying to solve problems where there are no customs yet to guide you. Maybe I can't plead Occam's razor; maybe I'm simply eccentric. But if I'm right about the acceleration of addictiveness, then this kind of lonely squirming to avoid it will increasingly be the fate of anyone who wants to get things done. We'll increasingly be defined by what we say no to.

Notes

[1] Could you restrict technological progress to areas where you wanted it? Only in a limited way, without becoming a police state. And even then your restrictions would have undesirable side effects. "Good" and "bad" technological progress aren't sharply differentiated, so you'd find you couldn't slow the latter without also slowing the former. And in any case, as Prohibition and the "war on drugs" show, bans often do more harm than good.

[2] Technology has always been accelerating. By Paleolithic standards, technology evolved at a blistering pace in the Neolithic period.

[3] Unless we mass produce social customs. I suspect the recent resurgence of evangelical Christianity in the US is partly a reaction to drugs. In desperation people reach for the sledgehammer; if their kids won't listen to them, maybe they'll listen to God. But that solution has broader consequences than just getting kids to say no to drugs. You end up saying no to science as well.

I worry we may be heading for a future in which only a few people plot their own itinerary through no-land, while everyone else books a package tour. Or worse still, has one booked for them by the government.

[4] People commonly use the word "procrastination" to describe what they do on the Internet. It seems to me too mild to describe what's happening as merely not-doing-work. We don't call it procrastination when someone gets drunk instead of working.

[5] Several people have told me they like the iPad because it lets them bring the Internet into situations where a laptop would be too conspicuous. In other words, it's a hip flask. (This is true of the iPhone too, of course, but this advantage isn't as obvious because it reads as a phone, and everyone's used to those.)

Thanks to Sam Altman, Patrick Collison, Jessica Livingston, and Robert Morris for reading drafts of this.

134 What We Look for in Founders

Want to start a startup? Get funded by Y Combinator.

October 2010

(I wrote this for Forbes, who asked me to write something about the qualities we look for in founders. In print they had to cut the last item because they didn't have room.)

1. Determination

This has turned out to be the most important quality in startup founders. We thought when we started Y Combinator that the most important quality would be intelligence. That's the myth in the Valley. And certainly you don't want founders to be stupid. But as long as you're over a certain threshold of intelligence, what matters most is determination. You're going to hit a lot of obstacles. You can't be the sort of person who gets demoralized easily.

Bill Clerico and Rich Aberman of WePay are a good example. They're doing a finance startup, which means endless negotiations with big, bureaucratic companies. When you're starting a startup that depends on deals with big companies to exist, it often feels like they're trying to ignore you out of existence. But when Bill Clerico starts calling you, you may as well do what he asks, because he is not going away.

2. Flexibility

You do not however want the sort of determination implied by phrases like "don't give up on your dreams." The world of startups is so unpredictable that you need to be able to modify your dreams on the fly. The best metaphor I've found for the combination of determination and flexibility you need is a running back. He's determined to get downfield, but at any given moment he may need to go sideways or even backwards to get there.

The current record holder for flexibility may be Daniel Gross of Greplin. He applied to YC with some bad ecommerce idea. We told him we'd fund him if he did something else. He thought for a second, and said ok. He then went through two more ideas before settling on Greplin. He'd only been working on it for a couple days when he presented to investors at Demo Day, but he got a lot of interest. He always seems to land on his feet.

3. Imagination

Intelligence does matter a lot of course. It seems like the type that matters most is imagination. It's not so important to be able to solve predefined problems quickly as to be able to come up with surprising new ideas. In the startup world, most good ideas seem bad initially. If they were obviously good, someone would already be doing them. So you need the kind of intelligence that produces ideas with just the right level of craziness.

Airbnb is that kind of idea. In fact, when we funded Airbnb, we thought it was too crazy. We couldn't believe large numbers of people would want to stay in other people's places. We funded them because we liked the founders so much. As soon as we heard they'd been supporting themselves by selling Obama and McCain branded breakfast cereal, they were in. And it turned out the idea was on the right side of crazy after all.

4. Naughtiness

Though the most successful founders are usually good people, they tend to have a piratical gleam in their eye. They're not Goody Two-Shoes type good. Morally, they care about getting the big questions right, but not about observing proprieties. That's why I'd use the word naughty rather than evil. They delight in breaking rules, but not rules that matter. This quality may be redundant though; it may be implied by imagination.

Sam Altman of Loopt is one of the most successful alumni, so we asked him what question we could put on the Y Combinator application that would help us discover more people like him. He said to ask about a time when they'd hacked something to their advantage--hacked in the sense of beating the system, not breaking into computers. It has become one of the questions we pay most attention to when judging applications.

5. Friendship

Empirically it seems to be hard to start a startup with just one founder. Most of the big successes have two or three. And the relationship between the founders has to be strong. They must genuinely like one another, and work well together. Startups do to the relationship between the founders what a dog does to a sock: if it can be pulled apart, it will be.

Emmett Shear and Justin Kan of Justin.tv are a good example of close friends who work well together. They've known each other since second grade. They can practically read one another's minds. I'm sure they argue, like all founders, but I have never once sensed any unresolved tension between them.

Thanks to Jessica Livingston and Chris Steiner for reading drafts of this.

141 Schlep Blindness

Want to start a startup? Get funded by Y Combinator.

January 2012

There are great startup ideas lying around unexploited right under our noses. One reason we don't see them is a phenomenon I call *schlep blindness*. Schlep was originally a Yiddish word but has passed into general use in the US. It means a tedious, unpleasant task.

No one likes schleps, but hackers especially dislike them. Most hackers who start startups wish they could do it by just writing some clever software, putting it on a server somewhere, and watching the money roll in--without ever having to talk to users, or negotiate with other companies, or deal with other people's broken code. Maybe that's possible, but I haven't seen it.

One of the many things we do at Y Combinator is teach hackers about the inevitability of schleps. No, you can't start a startup by just writing code. I remember going through this realization myself. There was a point in 1995 when I was still trying to convince myself I could start a company by just writing code. But I soon learned from experience that schleps are not merely inevitable, but pretty much what business consists of. A company is defined by the schleps it will undertake. And schleps should be dealt with the same way you'd deal with a cold swimming pool: just jump in. Which is not to say you should seek out unpleasant work per se, but that you should never shrink from it if it's on the path to something great.

The most dangerous thing about our dislike of schleps is that much of it is unconscious. Your unconscious won't even let you see ideas that involve painful schleps. That's schlep blindness.

The phenomenon isn't limited to startups. Most people don't consciously decide not to be in as good physical shape as Olympic athletes, for example. Their unconscious mind decides for them, shrinking from the work involved.

The most striking example I know of schlep blindness is Stripe, or rather Stripe's idea. For over a decade, every hacker who'd ever had to process payments online knew how painful the experience was. Thousands of people must have known about this problem. And yet when they started startups, they decided to build recipe sites, or aggregators for local events. Why? Why work on problems few care much about and no one will pay for, when you could fix one of the most important components of the world's infrastructure? Because schlep blindness prevented people from even considering the idea of fixing payments.

Probably no one who applied to Y Combinator to work on a recipe site began by asking "should we fix payments, or build a recipe site?" and chose the recipe site. Though the idea of fixing payments was right there in plain sight, they never saw it, because their unconscious mind shrank from the complications involved. You'd have to make deals with banks. How do you do that? Plus you're moving money, so you're going to have to deal with fraud, and people trying to break into your servers. Plus there are probably all sorts of regulations to comply with. It's a lot more intimidating to start a startup like this than a recipe site.

That scariness makes ambitious ideas doubly valuable. In addition to their intrinsic value, they're like undervalued stocks in the sense that there's less demand for them among founders. If you pick an ambitious idea, you'll have less competition, because everyone else will have been frightened off by the challenges involved. (This is also true of starting a startup generally.)

How do you overcome schlep blindness? Frankly, the most valuable antidote to schlep blindness is probably ignorance. Most successful founders would probably say that if they'd known when they were starting their company about the obstacles they'd have to overcome, they might never have started it. Maybe that's one reason the most successful startups of all so often have young founders.

In practice the founders grow with the problems. But no one seems able to foresee that, not even older, more experienced founders. So the reason younger founders have an advantage is that they make two mistakes that cancel each other out. They don't know how much they can grow, but they also don't know how much they'll need to. Older founders only make the first mistake.

Ignorance can't solve everything though. Some ideas so obviously entail alarming schleps that anyone can see them. How do you see ideas like that? The trick I recommend is to take yourself out of the picture. Instead of asking "what problem should I solve?" ask "what problem do I wish someone else would solve for me?" If someone who had to process payments before Stripe had tried asking that, Stripe would have been one of the first things they wished for.

It's too late now to be Stripe, but there's plenty still broken in the world, if you know how to see it.

Thanks to Sam Altman, Paul Buchheit, Patrick Collison, Aaron Iba, Jessica Livingston, Emmett Shear, and Harj Taggar for reading drafts of this.

146 Writing and Speaking

March 2012

I'm not a very good speaker. I say "um" a lot. Sometimes I have to pause when I lose my train of thought. I wish I were a better speaker. But I don't wish I were a better speaker like I wish I were a better writer. What I really want is to have good ideas, and that's a much bigger part of being a good writer than being a good speaker.

Having good ideas is most of writing well. If you know what you're talking about, you can say it in the plainest words and you'll be perceived as having a good style. With speaking it's the opposite: having good ideas is an alarmingly small component of being a good speaker.

I first noticed this at a conference several years ago. There was another speaker who was much better than me. He had all of us roaring with laughter. I seemed awkward and halting by comparison. Afterward I put my talk online like I usually do. As I was doing it I tried to imagine what a transcript of the other guy's talk would be like, and it was only then I realized he hadn't said very much.

Maybe this would have been obvious to someone who knew more about speaking, but it was a revelation to me how much less ideas mattered in speaking than writing. 1

A few years later I heard a talk by someone who was not merely a better speaker than me, but a famous speaker. Boy was he good. So I decided I'd pay close attention to what he said, to learn how he did it. After about ten sentences I found myself thinking "I don't want to be a good speaker."

Being a really good speaker is not merely orthogonal to having good ideas, but in many ways pushes you in the opposite direction. For example, when I give a talk, I usually write it out beforehand. I know that's a mistake; I know delivering a prewritten talk makes it harder to engage with an audience. The way to get the attention of an audience is to give them *your* full attention, and when you're delivering a prewritten talk, your attention is always divided between the audience and the talk -- even if you've memorized it. If you want to engage an audience, it's better to start with no more than an outline of what you want to say and ad lib the individual sentences. But if you do that, you might spend no more time thinking about each sentence than it takes to say it. 2 Occasionally the stimulation of talking to a live audience makes you think of new things, but in general this is not going to generate ideas as well as writing does, where you can spend as long on each sentence as you want.

If you rehearse a prewritten speech enough, you can get asymptotically close to the sort of engagement you get when speaking ad lib. Actors do. But here again there's a tradeoff between smoothness and ideas. All the time you spend practicing a talk, you could instead spend making it better. Actors don't face that temptation, except in the rare cases where they've written the script, but any speaker does. Before I give a talk I can usually be found sitting in a corner somewhere with a copy printed out on paper, trying to rehearse it in my head. But I always end up spending most of the time rewriting it instead. Every talk I give ends up being given from a manuscript full of things crossed out and rewritten. Which of course makes me um even more, because I haven't had any time to practice the new bits. 3

Depending on your audience, there are even worse tradeoffs than these. Audiences like to be flattered; they like jokes; they like to be swept off their feet by a vigorous stream of words. As you decrease the intelligence of the audience, being a good speaker is increasingly a matter of being a good bullshitter. That's true in writing too of course, but the descent is steeper with talks. Any given person is dumber as a member of an audience than as a reader. Just as a speaker ad libbing can only spend as long thinking about each sentence as it takes to say it, a person hearing a talk can only spend as long thinking about each sentence as it takes to hear it. Plus people in an audience are always affected by the reactions of those around them, and the reactions that spread from person to person in an audience are disproportionately the more brutish sort, just as low notes travel through walls better than high ones. Every audience is an incipient mob, and a good speaker uses that. Part of the reason I laughed so much at the

talk by the good speaker at that conference was that everyone else did. 4

So are talks useless? They're certainly inferior to the written word as a source of ideas. But that's not all talks are good for. When I go to a talk, it's usually because I'm interested in the speaker. Listening to a talk is the closest most of us can get to having a conversation with someone like the president, who doesn't have time to meet individually with all the people who want to meet him.

Talks are also good at motivating me to do things. It's probably no coincidence that so many famous speakers are described as motivational speakers. That may be what public speaking is really for. It's probably what it was originally for. The emotional reactions you can elicit with a talk can be a powerful force. I wish I could say that this force was more often used for good than ill, but I'm not sure.

Notes

[1] I'm not talking here about academic talks, which are a different type of thing. While the audience at an academic talk might appreciate a joke, they will (or at least should) make a conscious effort to see what new ideas you're presenting.

[2] That's the lower bound. In practice you can often do better, because talks are usually about things you've written or talked about before, and when you ad lib, you end up reproducing some of those sentences. Like early medieval architecture, impromptu talks are made of spolia. Which feels a bit dishonest, incidentally, because you have to deliver these sentences as if you'd just thought of them.

[3] Robert Morris points out that there is a way in which practicing talks makes them better: reading a talk out loud can expose awkward parts. I agree and in fact I read most things I write out loud at least once for that reason.

[4] For sufficiently small audiences, it may not be true that being part of an audience makes people dumber. The real decline seems to set in when the audience gets too big for the talk to feel like a conversation -- maybe around 10 people.

Thanks to Sam Altman and Robert Morris for reading drafts of this.

147 The Top of My Todo List

April 2012

A palliative care nurse called Bronnie Ware made a list of the biggest regrets of the dying. Her list seems plausible. I could see myself -- *can* see myself -- making at least 4 of these 5 mistakes.

If you had to compress them into a single piece of advice, it might be: don't be a cog. The 5 regrets paint a portrait of post-industrial man, who shrinks himself into a shape that fits his circumstances, then turns dutifully till he stops.

The alarming thing is, the mistakes that produce these regrets are all errors of omission. You forget your dreams, ignore your family, suppress your feelings, neglect your friends, and forget to be happy. Errors of omission are a particularly dangerous type of mistake, because you make them by default.

I would like to avoid making these mistakes. But how do you avoid mistakes you make by default? Ideally you transform your life so it has other defaults. But it may not be possible to do that completely. As long as these mistakes happen by default, you probably have to be reminded not to make them. So I inverted the 5 regrets, yielding a list of 5 commands

Don't ignore your dreams; don't work too much; say what you think;
cultivate > friendships; be happy.

which I then put at the top of the file I use as a todo list.

Japanese Translation

148 Black Swan Farming

Want to start a startup? Get funded by Y Combinator.

September 2012

I've done several types of work over the years but I don't know another as counterintuitive as startup investing.

The two most important things to understand about startup investing, as a business, are (1) that effectively all the returns are concentrated in a few big winners, and (2) that the best ideas look initially like bad ideas.

The first rule I knew intellectually, but didn't really grasp till it happened to us. The total value of the companies we've funded is around 10 billion, give or take a few. But just two companies, Dropbox and Airbnb, account for about three quarters of it.

In startups, the big winners are big to a degree that violates our expectations about variation. I don't know whether these expectations are innate or learned, but whatever the cause, we are just not prepared for the 1000x variation in outcomes that one finds in startup investing.

That yields all sorts of strange consequences. For example, in purely financial terms, there is probably at most one company in each YC batch that will have a significant effect on our returns, and the rest are just a cost of doing business. I haven't really assimilated that fact, partly because it's so counterintuitive, and

partly because we're not doing this just for financial reasons; YC would be a pretty lonely place if we only had one company per batch. And yet it's true.

To succeed in a domain that violates your intuitions, you need to be able to turn them off the way a pilot does when flying through clouds. ² You need to do what you know intellectually to be right, even though it feels wrong.

It's a constant battle for us. It's hard to make ourselves take enough risks. When you interview a startup and think "they seem likely to succeed," it's hard not to fund them. And yet, financially at least, there is only one kind of success: they're either going to be one of the really big winners or not, and if not it doesn't matter whether you fund them, because even if they succeed the effect on your returns will be insignificant. In the same day of interviews you might meet some smart 19 year olds who aren't even sure what they want to work on. Their chances of succeeding seem small. But again, it's not their chances of succeeding that matter but their chances of succeeding really big. The probability that any group will succeed really big is microscopically small, but the probability that those 19 year olds will might be higher than that of the other, safer group.

The probability that a startup will make it big is not simply a constant fraction of the probability that they will succeed at all. If it were, you could fund everyone who seemed likely to succeed at all, and you'd get that fraction of big hits. Unfortunately picking winners is harder than that. You have to ignore the elephant in front of you, the likelihood they'll succeed, and focus instead on the separate and almost invisibly intangible question of whether they'll succeed really big.

Harder

That's made harder by the fact that the best startup ideas seem at first like bad ideas. I've written about this before: if a good idea were obviously good, someone else would already have done it. So the most successful founders tend to work on ideas that few beside them realize are good. Which is not that far from a description of insanity, till you reach the point where you see results.

The first time Peter Thiel spoke at YC he drew a Venn diagram that illustrates the situation perfectly. He drew two intersecting circles, one labelled "seems like a bad idea" and the other "is a good idea." The intersection is the sweet spot for startups.

This concept is a simple one and yet seeing it as a Venn diagram is illuminating. It reminds you that there is an intersection--that there are good ideas that seem bad. It also reminds you that the vast majority of ideas that seem bad are bad.

The fact that the best ideas seem like bad ideas makes it even harder to recognize the big winners. It means the probability of a startup making it really big is not merely not a constant fraction of the probability that it will succeed, but that the startups with a high probability of the former will seem to have a disproportionately low probability of the latter.

History tends to get rewritten by big successes, so that in retrospect it seems obvious they were going to make it big. For that reason one of my most valuable memories is how lame Facebook sounded to me when I first heard about it. A site for college students to waste time? It seemed the perfect bad idea: a site (1) for a niche market (2) with no money (3) to do something that didn't matter.

One could have described Microsoft and Apple in exactly the same terms. 3

Harder Still

Wait, it gets worse. You not only have to solve this hard problem, but you have to do it with no indication of whether you're succeeding. When you pick a big winner, you won't know it for two years.

Meanwhile, the one thing you *can* measure is dangerously misleading. The one thing we can track precisely is how well the startups in each batch do at fundraising after Demo Day. But we know that's the wrong metric. There's no correlation between the percentage of startups that raise money and the metric that does matter financially, whether that batch of startups contains a big winner or not.

Except an inverse one. That's the scary thing: fundraising is not merely a useless metric, but positively misleading. We're in a business where we need to pick unpromising-looking outliers, and the huge scale of the successes means we can afford to spread our net very widely. The big winners could generate 10,000x returns. That means for each big winner we could pick a thousand companies that returned nothing and still end up 10x ahead.

If we ever got to the point where 100% of the startups we funded were able to raise money after Demo Day, it would almost certainly mean we were being too conservative. 4

It takes a conscious effort not to do that too. After 15 cycles of preparing startups for investors and then watching how they do, I can now look at a group we're interviewing through Demo Day investors' eyes. But those are the wrong eyes to look through!

We can afford to take at least 10x as much risk as Demo Day investors. And since risk is usually proportionate to reward, if you can afford to take more risk you should. What would it mean to take 10x more risk than Demo Day investors? We'd have to be willing to fund 10x more startups than they would. Which means that even if we're generous to ourselves and assume that YC can on average triple a startup's expected value, we'd be taking the right amount of risk if only 30% of the startups were able to raise significant funding after Demo Day.

I don't know what fraction of them currently raise more after Demo Day. I deliberately avoid calculating that number, because if you start measuring something you start optimizing it, and I know it's the wrong thing to optimize. 5 But the percentage is certainly way over 30%. And frankly the thought of a

30% success rate at fundraising makes my stomach clench. A Demo Day where only 30% of the startups were fundable would be a shambles. Everyone would agree that YC had jumped the shark. We ourselves would feel that YC had jumped the shark. And yet we'd all be wrong.

For better or worse that's never going to be more than a thought experiment. We could never stand it. How about that for counterintuitive? I can lay out what I know to be the right thing to do, and still not do it. I can make up all sorts of plausible justifications. It would hurt YC's brand (at least among the innumerate) if we invested in huge numbers of risky startups that flamed out. It might dilute the value of the alumni network. Perhaps most convincingly, it would be demoralizing for us to be up to our chins in failure all the time. But I know the real reason we're so conservative is that we just haven't assimilated the fact of 1000x variation in returns.

We'll probably never be able to bring ourselves to take risks proportionate to the returns in this business. The best we can hope for is that when we interview a group and find ourselves thinking "they seem like good founders, but what are investors going to think of this crazy idea?" we'll continue to be able to say "who cares what investors think?" That's what we thought about Airbnb, and if we want to fund more Airbnbs we have to stay good at thinking it.

Notes

[1] I'm not saying that the big winners are all that matters, just that they're all that matters financially for investors. Since we're not doing YC mainly for financial reasons, the big winners aren't all that matters to us. We're delighted to have funded Reddit, for example. Even though we made comparatively little from it, Reddit has had a big effect on the world, and it introduced us to Steve Huffman and Alexis Ohanian, both of whom have become good friends.

Nor do we push founders to try to become one of the big winners if they don't want to. We didn't "swing for the fences" in our own startup (Viaweb, which was acquired for \$50 million), and it would feel pretty bogus to press founders to do something we didn't do. Our rule is that it's up to the founders. Some want to take over the world, and some just want that first few million. But we invest in so many companies that we don't have to sweat any one outcome. In fact, we don't have to sweat whether startups have exits at all. The biggest exits are the only ones that matter financially, and those are guaranteed in the sense that if a company becomes big enough, a market for its shares will inevitably arise. Since the remaining outcomes don't have a significant effect on returns, it's cool with us if the founders want to sell early for a small amount, or grow slowly and never sell (i.e. become a so-called lifestyle business), or even shut the company down. We're sometimes disappointed when a startup we had high hopes for doesn't do well, but this disappointment is mostly the ordinary variety that anyone feels when that happens.

[2] Without visual cues (e.g. the horizon) you can't distinguish between gravity and acceleration. Which means if you're flying through clouds you can't tell

what the attitude of the aircraft is. You could feel like you're flying straight and level while in fact you're descending in a spiral. The solution is to ignore what your body is telling you and listen only to your instruments. But it turns out to be very hard to ignore what your body is telling you. Every pilot knows about this problem and yet it is still a leading cause of accidents.

[3] Not all big hits follow this pattern though. The reason Google seemed a bad idea was that there were already lots of search engines and there didn't seem to be room for another.

[4] A startup's success at fundraising is a function of two things: what they're selling and how good they are at selling it. And while we can teach startups a lot about how to appeal to investors, even the most convincing pitch can't sell an idea that investors don't like. I was genuinely worried that Airbnb, for example, would not be able to raise money after Demo Day. I couldn't convince Fred Wilson to fund them. They might not have raised money at all but for the coincidence that Greg McAdoo, our contact at Sequoia, was one of a handful of VCs who understood the vacation rental business, having spent much of the previous two years investigating it.

[5] I calculated it once for the last batch before a consortium of investors started offering investment automatically to every startup we funded, summer 2010. At the time it was 94% (33 of 35 companies that tried to raise money succeeded, and one didn't try because they were already profitable). Presumably it's lower now because of that investment; in the old days it was raise after Demo Day or die.

Thanks to Sam Altman, Paul Buchheit, Patrick Collison, Jessica Livingston, Geoff Ralston, and Harj Taggar for reading drafts of this.

164 What Doesn't Seem Like Work?

January 2015

My father is a mathematician. For most of my childhood he worked for Westinghouse, modelling nuclear reactors.

He was one of those lucky people who know early on what they want to do. When you talk to him about his childhood, there's a clear watershed at about age 12, when he "got interested in maths."

He grew up in the small Welsh seacoast town of Pwllheli. As we retraced his walk to school on Google Street View, he said that it had been nice growing up in the country.

"Didn't it get boring when you got to be about 15?" I asked.

"No," he said, "by then I was interested in maths."

In another conversation he told me that what he really liked was solving problems. To me the exercises at the end of each chapter in a math textbook represent work, or at best a way to reinforce what you learned in that chapter. To him the problems were the reward. The text of each chapter was just some advice about solving them. He said that as soon as he got a new textbook he'd immediately work out all the problems -- to the slight annoyance of his teacher, since the class was supposed to work through the book gradually.

Few people know so early or so certainly what they want to work on. But talking to my father reminded me of a heuristic the rest of us can use. If something that seems like work to other people doesn't seem like work to you, that's something you're well suited for. For example, a lot of programmers I know, including me, actually like debugging. It's not something people tend to volunteer; one likes it the way one likes popping zits. But you may have to like debugging to like programming, considering the degree to which programming consists of it.

The stranger your tastes seem to other people, the stronger evidence they probably are of what you should do. When I was in college I used to write papers for my friends. It was quite interesting to write a paper for a class I wasn't taking. Plus they were always so relieved.

It seemed curious that the same task could be painful to one person and pleasant to another, but I didn't realize at the time what this imbalance implied, because I wasn't looking for it. I didn't realize how hard it can be to decide what you should work on, and that you sometimes have to figure it out from subtle clues, like a detective solving a case in a mystery novel. So I bet it would help a lot of people to ask themselves about this explicitly. What seems like work to other people that doesn't seem like work to you?

Thanks to Sam Altman, Trevor Blackwell, Jessica Livingston, Robert Morris, and my father for reading drafts of this.

Robert Morris: All About Programming

French Translation

169 Default Alive or Default Dead?

October 2015

When I talk to a startup that's been operating for more than 8 or 9 months, the first thing I want to know is almost always the same. Assuming their expenses remain constant and their revenue growth is what it has been over the last several months, do they make it to profitability on the money they have left? Or to put it more dramatically, by default do they live or die?

The startling thing is how often the founders themselves don't know. Half the founders I talk to don't know whether they're default alive or default dead.

If you're among that number, Trevor Blackwell has made a handy *calculator* you can use to find out.

The reason I want to know first whether a startup is default alive or default dead is that the rest of the conversation depends on the answer. If the company is default alive, we can talk about ambitious new things they could do. If it's default dead, we probably need to talk about how to save it. We know the current trajectory ends badly. How can they get off that trajectory?

Why do so few founders know whether they're default alive or default dead? Mainly, I think, because they're not used to asking that. It's not a question that makes sense to ask early on, any more than it makes sense to ask a 3 year old how he plans to support himself. But as the company grows older, the question switches from meaningless to critical. That kind of switch often takes people by surprise.

I propose the following solution: instead of starting to ask too late whether you're default alive or default dead, start asking too early. It's hard to say precisely when the question switches polarity. But it's probably not that dangerous to start worrying too early that you're default dead, whereas it's very dangerous to start worrying too late.

The reason is a phenomenon I wrote about earlier: the fatal pinch. The fatal pinch is default dead + slow growth + not enough time to fix it. And the way founders end up in it is by not realizing that's where they're headed.

There is another reason founders don't ask themselves whether they're default alive or default dead: they assume it will be easy to raise more money. But that assumption is often false, and worse still, the more you depend on it, the falser it becomes.

Maybe it will help to separate facts from hopes. Instead of thinking of the future with vague optimism, explicitly separate the components. Say "We're default dead, but we're counting on investors to save us." Maybe as you say that, it will set off the same alarms in your head that it does in mine. And if you set off the alarms sufficiently early, you may be able to avoid the fatal pinch.

It would be safe to be default dead if you could count on investors saving you. As a rule their interest is a function of growth. If you have steep revenue growth, say over 5x a year, you can start to count on investors being interested even if you're not profitable. ¹ But investors are so fickle that you can never do more than start to count on them. Sometimes something about your business will spook investors even if your growth is great. So no matter how good your growth is, you can never safely treat fundraising as more than a plan A. You should always have a plan B as well: you should know (as in write down) precisely what you'll need to do to survive if you can't raise more money, and precisely when you'll have to switch to plan B if plan A isn't working.

In any case, growing fast versus operating cheaply is far from the sharp dichotomy many founders assume it to be. In practice there is surprisingly little connection between how much a startup spends and how fast it grows. When a startup grows fast, it's usually because the product hits a nerve, in the sense of hitting some big need straight on. When a startup spends a lot, it's usually because the product is expensive to develop or sell, or simply because they're wasteful.

If you're paying attention, you'll be asking at this point not just how to avoid the fatal pinch, but how to avoid being default dead. That one is easy: don't hire too fast. Hiring too fast is by far the biggest killer of startups that raise money. 2

Founders tell themselves they need to hire in order to grow. But most err on the side of overestimating this need rather than underestimating it. Why? Partly because there's so much work to do. Naive founders think that if they can just hire enough people, it will all get done. Partly because successful startups have lots of employees, so it seems like that's what one does in order to be successful. In fact the large staffs of successful startups are probably more the effect of growth than the cause. And partly because when founders have slow growth they don't want to face what is usually the real reason: the product is not appealing enough.

Plus founders who've just raised money are often encouraged to overhire by the VCs who funded them. Kill-or-cure strategies are optimal for VCs because they're protected by the portfolio effect. VCs want to blow you up, in one sense of the phrase or the other. But as a founder your incentives are different. You want above all to survive. 3

Here's a common way startups die. They make something moderately appealing and have decent initial growth. They raise their first round fairly easily, because the founders seem smart and the idea sounds plausible. But because the product is only moderately appealing, growth is ok but not great. The founders convince themselves that hiring a bunch of people is the way to boost growth. Their investors agree. But (because the product is only moderately appealing) the growth never comes. Now they're rapidly running out of runway. They hope further investment will save them. But because they have high expenses and slow growth, they're now unappealing to investors. They're unable to raise more, and the company dies.

What the company should have done is address the fundamental problem: that the product is only moderately appealing. Hiring people is rarely the way to fix that. More often than not it makes it harder. At this early stage, the product needs to evolve more than to be "built out," and that's usually easier with fewer people. 4

Asking whether you're default alive or default dead may save you from this. Maybe the alarm bells it sets off will counteract the forces that push you to overhire. Instead you'll be compelled to seek growth in other ways. For example, by *doing things that don't scale*, or by redesigning the product in the way only

founders can. And for many if not most startups, these paths to growth will be the ones that actually work.

Airbnb waited 4 months after raising money at the end of Y Combinator before they hired their first employee. In the meantime the founders were terribly overworked. But they were overworked evolving Airbnb into the astonishingly successful organism it is now.

Notes

[1] Steep usage growth will also interest investors. Revenue will ultimately be a constant multiple of usage, so $x\%$ usage growth predicts $x\%$ revenue growth. But in practice investors discount merely predicted revenue, so if you're measuring usage you need a higher growth rate to impress investors.

[2] Startups that don't raise money are saved from hiring too fast because they can't afford to. But that doesn't mean you should avoid raising money in order to avoid this problem, any more than that total abstinence is the only way to avoid becoming an alcoholic.

[3] I would not be surprised if VCs' tendency to push founders to overhire is not even in their own interest. They don't know how many of the companies that get killed by overspending might have done well if they'd survived. My guess is a significant number.

[4] After reading a draft, Sam Altman wrote:

"I think you should make the hiring point more strongly. I think it's roughly correct to say that YC's most successful companies have never been the fastest to hire, and one of the marks of a great founder is being able to resist this urge."

Paul Buchheit adds:

"A related problem that I see a lot is premature scaling--founders take a small business that isn't really working (bad unit economics, typically) and then scale it up because they want impressive growth numbers. This is similar to over-hiring in that it makes the business much harder to fix once it's big, plus they are bleeding cash really fast."

Thanks to Sam Altman, Paul Buchheit, Joe Gebbia, Jessica Livingston, and Geoff Ralston for reading drafts of this.

170 Write Like You Talk

October 2015

Here's a simple trick for getting more people to read what you write: write in spoken language.

Something comes over most people when they start writing. They write in a different language than they'd use if they were talking to a friend. The sentence structure and even the words are different. No one uses "pen" as a verb in spoken English. You'd feel like an idiot using "pen" instead of "write" in a conversation with a friend.

The last straw for me was a sentence I read a couple days ago:

The mercurial Spaniard himself declared: "After Altamira, all is decadence."

It's from Neil Oliver's *A History of Ancient Britain*. I feel bad making an example of this book, because it's no worse than lots of others. But just imagine calling Picasso "the mercurial Spaniard" when talking to a friend. Even one sentence of this would raise eyebrows in conversation. And yet people write whole books of it.

Ok, so written and spoken language are different. Does that make written language worse?

If you want people to read and understand what you write, yes. Written language is more complex, which makes it more work to read. It's also more formal and distant, which gives the reader's attention permission to drift. But perhaps worst of all, the complex sentences and fancy words give you, the writer, the false impression that you're saying more than you actually are.

You don't need complex sentences to express complex ideas. When specialists in some abstruse topic talk to one another about ideas in their field, they don't use sentences any more complex than they do when talking about what to have for lunch. They use different words, certainly. But even those they use no more than necessary. And in my experience, the harder the subject, the more informally experts speak. Partly, I think, because they have less to prove, and partly because the harder the ideas you're talking about, the less you can afford to let language get in the way.

Informal language is the athletic clothing of ideas.

I'm not saying spoken language always works best. Poetry is as much music as text, so you can say things you wouldn't say in conversation. And there are a handful of writers who can get away with using fancy language in prose. And then of course there are cases where writers don't want to make it easy to understand what they're saying--in corporate announcements of bad news, for example, or at the more *bogus* end of the humanities. But for nearly everyone else, spoken language is better.

It seems to be hard for most people to write in spoken language. So perhaps the best solution is to write your first draft the way you usually would, then afterward look at each sentence and ask "Is this the way I'd say this if I were talking to a friend?" If it isn't, imagine what you would say, and use that instead.

After a while this filter will start to operate as you write. When you write something you wouldn't say, you'll hear the clank as it hits the page.

Before I publish a new essay, I read it out loud and fix everything that doesn't sound like conversation. I even fix bits that are phonetically awkward; I don't know if that's necessary, but it doesn't cost much.

This trick may not always be enough. I've seen writing so far removed from spoken language that it couldn't be fixed sentence by sentence. For cases like that there's a more drastic solution. After writing the first draft, try explaining to a friend what you just wrote. Then replace the draft with what you said to your friend.

People often tell me how much my essays sound like me talking. The fact that this seems worthy of comment shows how rarely people manage to write in spoken language. Otherwise everyone's writing would sound like them talking.

If you simply manage to write in spoken language, you'll be ahead of 95% of writers. And it's so easy to do: just don't let a sentence through unless it's the way you'd say it to a friend.

Thanks to Patrick Collison and Jessica Livingston for reading drafts of this.

Japanese Translation Arabic Translation

175 Life is Short

January 2016

Life is short, as everyone knows. When I was a kid I used to wonder about this. Is life actually short, or are we really complaining about its finiteness? Would we be just as likely to feel life was short if we lived 10 times as long?

Since there didn't seem any way to answer this question, I stopped wondering about it. Then I had kids. That gave me a way to answer the question, and the answer is that life actually is short.

Having kids showed me how to convert a continuous quantity, time, into discrete quantities. You only get 52 weekends with your 2 year old. If Christmas-as-magic lasts from say ages 3 to 10, you only get to watch your child experience it 8 times. And while it's impossible to say what is a lot or a little of a continuous quantity like time, 8 is not a lot of something. If you had a handful of 8 peanuts, or a shelf of 8 books to choose from, the quantity would definitely seem limited, no matter what your lifespan was.

Ok, so life actually is short. Does it make any difference to know that?

It has for me. It means arguments of the form "Life is too short for x" have great force. It's not just a figure of speech to say that life is too short for something.

It's not just a synonym for annoying. If you find yourself thinking that life is too short for something, you should try to eliminate it if you can.

When I ask myself what I've found life is too short for, the word that pops into my head is "bullshit." I realize that answer is somewhat tautological. It's almost the definition of bullshit that it's the stuff that life is too short for. And yet bullshit does have a distinctive character. There's something fake about it. It's the junk food of experience. 1

If you ask yourself what you spend your time on that's bullshit, you probably already know the answer. Unnecessary meetings, pointless disputes, bureaucracy, posturing, dealing with other people's mistakes, traffic jams, addictive but unrewarding pastimes.

There are two ways this kind of thing gets into your life: it's either forced on you, or it tricks you. To some extent you have to put up with the bullshit forced on you by circumstances. You need to make money, and making money consists mostly of errands. Indeed, the law of supply and demand ensures that: the more rewarding some kind of work is, the cheaper people will do it. It may be that less bullshit is forced on you than you think, though. There has always been a stream of people who opt out of the default grind and go live somewhere where opportunities are fewer in the conventional sense, but life feels more authentic. This could become more common.

You can do it on a smaller scale without moving. The amount of time you have to spend on bullshit varies between employers. Most large organizations (and many small ones) are steeped in it. But if you consciously prioritize bullshit avoidance over other factors like money and prestige, you can probably find employers that will waste less of your time.

If you're a freelancer or a small company, you can do this at the level of individual customers. If you fire or avoid toxic customers, you can decrease the amount of bullshit in your life by more than you decrease your income.

But while some amount of bullshit is inevitably forced on you, the bullshit that sneaks into your life by tricking you is no one's fault but your own. And yet the bullshit you choose may be harder to eliminate than the bullshit that's forced on you. Things that lure you into wasting your time have to be really good at tricking you. An example that will be familiar to a lot of people is arguing online. When someone contradicts you, they're in a sense attacking you. Sometimes pretty overtly. Your instinct when attacked is to defend yourself. But like a lot of instincts, this one wasn't designed for the world we now live in. Counterintuitive as it feels, it's better most of the time not to defend yourself. Otherwise these people are literally taking your life. 2

Arguing online is only incidentally addictive. There are more dangerous things than that. As I've written before, one byproduct of technical progress is that things we like tend to become *more addictive*. Which means we will increasingly have to make a conscious effort to avoid addictions — to stand outside ourselves

and ask "is this how I want to be spending my time?"

As well as avoiding bullshit, one should actively seek out things that matter. But different things matter to different people, and most have to learn what matters to them. A few are lucky and realize early on that they love math or taking care of animals or writing, and then figure out a way to spend a lot of time doing it. But most people start out with a life that's a mix of things that matter and things that don't, and only gradually learn to distinguish between them.

For the young especially, much of this confusion is induced by the artificial situations they find themselves in. In middle school and high school, what the other kids think of you seems the most important thing in the world. But when you ask adults what they got wrong at that age, nearly all say they cared too much what other kids thought of them.

One heuristic for distinguishing stuff that matters is to ask yourself whether you'll care about it in the future. Fake stuff that matters usually has a sharp peak of seeming to matter. That's how it tricks you. The area under the curve is small, but its shape jabs into your consciousness like a pin.

The things that matter aren't necessarily the ones people would call "important." Having coffee with a friend matters. You won't feel later like that was a waste of time.

One great thing about having small children is that they make you spend time on things that matter: them. They grab your sleeve as you're staring at your phone and say "will you play with me?" And odds are that is in fact the bullshit-minimizing option.

If life is short, we should expect its shortness to take us by surprise. And that is just what tends to happen. You take things for granted, and then they're gone. You think you can always write that book, or climb that mountain, or whatever, and then you realize the window has closed. The saddest windows close when other people die. Their lives are short too. After my mother died, I wished I'd spent more time with her. I lived as if she'd always be there. And in her typical quiet way she encouraged that illusion. But an illusion it was. I think a lot of people make the same mistake I did.

The usual way to avoid being taken by surprise by something is to be consciously aware of it. Back when life was more precarious, people used to be aware of death to a degree that would now seem a bit morbid. I'm not sure why, but it doesn't seem the right answer to be constantly reminding oneself of the grim reaper hovering at everyone's shoulder. Perhaps a better solution is to look at the problem from the other end. Cultivate a habit of impatience about the things you most want to do. Don't wait before climbing that mountain or writing that book or visiting your mother. You don't need to be constantly reminding yourself why you shouldn't wait. Just don't wait.

I can think of two more things one does when one doesn't have much of something: try to get more of it, and savor what one has. Both make sense here.

How you live affects how long you live. Most people could do better. Me among them.

But you can probably get even more effect by paying closer attention to the time you have. It's easy to let the days rush by. The "flow" that imaginative people love so much has a darker cousin that prevents you from pausing to savor life amid the daily slurry of errands and alarms. One of the most striking things I've read was not in a book, but the title of one: James Salter's *Burning the Days*.

It is possible to slow time somewhat. I've gotten better at it. Kids help. When you have small children, there are a lot of moments so perfect that you can't help noticing.

It does help too to feel that you've squeezed everything out of some experience. The reason I'm sad about my mother is not just that I miss her but that I think of all the things we could have done that we didn't. My oldest son will be 7 soon. And while I miss the 3 year old version of him, I at least don't have any regrets over what might have been. We had the best time a daddy and a 3 year old ever had.

Relentlessly prune bullshit, don't wait to do things that matter, and savor the time you have. That's what you do when life is short.

Notes

[1] At first I didn't like it that the word that came to mind was one that had other meanings. But then I realized the other meanings are fairly closely related. Bullshit in the sense of things you waste your time on is a lot like intellectual bullshit.

[2] I chose this example deliberately as a note to self. I get attacked a lot online. People tell the craziest lies about me. And I have so far done a pretty mediocre job of suppressing the natural human inclination to say "Hey, that's not true!"

Thanks to Jessica Livingston and Geoff Ralston for reading drafts of this.

Korean Translation Japanese Translation

Chinese Translation

182 The Lesson to Unlearn

December 2019

The most damaging thing you learned in school wasn't something you learned in any specific class. It was learning to get good grades.

When I was in college, a particularly earnest philosophy grad student once told me that he never cared what grade he got in a class, only what he learned in

it. This stuck in my mind because it was the only time I ever heard anyone say such a thing.

For me, as for most students, the measurement of what I was learning completely dominated actual learning in college. I was fairly earnest; I was genuinely interested in most of the classes I took, and I worked hard. And yet I worked by far the hardest when I was studying for a test.

In theory, tests are merely what their name implies: tests of what you've learned in the class. In theory you shouldn't have to prepare for a test in a class any more than you have to prepare for a blood test. In theory you learn from taking the class, from going to the lectures and doing the reading and/or assignments, and the test that comes afterward merely measures how well you learned.

In practice, as almost everyone reading this will know, things are so different that hearing this explanation of how classes and tests are meant to work is like hearing the etymology of a word whose meaning has changed completely. In practice, the phrase "studying for a test" was almost redundant, because that was when one really studied. The difference between diligent and slack students was that the former studied hard for tests and the latter didn't. No one was pulling all-nighters two weeks into the semester.

Even though I was a diligent student, almost all the work I did in school was aimed at getting a good grade on something.

To many people, it would seem strange that the preceding sentence has a "though" in it. Aren't I merely stating a tautology? Isn't that what a diligent student is, a straight-A student? That's how deeply the conflation of learning with grades has infused our culture.

Is it so bad if learning is conflated with grades? Yes, it is bad. And it wasn't till decades after college, when I was running Y Combinator, that I realized how bad it is.

I knew of course when I was a student that studying for a test is far from identical with actual learning. At the very least, you don't retain knowledge you cram into your head the night before an exam. But the problem is worse than that. The real problem is that most tests don't come close to measuring what they're supposed to.

If tests truly were tests of learning, things wouldn't be so bad. Getting good grades and learning would converge, just a little late. The problem is that nearly all tests given to students are terribly hackable. Most people who've gotten good grades know this, and know it so well they've ceased even to question it. You'll see when you realize how naive it sounds to act otherwise.

Suppose you're taking a class on medieval history and the final exam is coming up. The final exam is supposed to be a test of your knowledge of medieval history, right? So if you have a couple days between now and the exam, surely the best way to spend the time, if you want to do well on the exam, is to read

the best books you can find about medieval history. Then you'll know a lot about it, and do well on the exam.

No, no, no, experienced students are saying to themselves. If you merely read good books on medieval history, most of the stuff you learned wouldn't be on the test. It's not good books you want to read, but the lecture notes and assigned reading in this class. And even most of that you can ignore, because you only have to worry about the sort of thing that could turn up as a test question. You're looking for sharply-defined chunks of information. If one of the assigned readings has an interesting digression on some subtle point, you can safely ignore that, because it's not the sort of thing that could be turned into a test question. But if the professor tells you that there were three underlying causes of the Schism of 1378, or three main consequences of the Black Death, you'd better know them. And whether they were in fact the causes or consequences is beside the point. For the purposes of this class they are.

At a university there are often copies of old exams floating around, and these narrow still further what you have to learn. As well as learning what kind of questions this professor asks, you'll often get actual exam questions. Many professors re-use them. After teaching a class for 10 years, it would be hard not to, at least inadvertently.

In some classes, your professor will have had some sort of political axe to grind, and if so you'll have to grind it too. The need for this varies. In classes in math or the hard sciences or engineering it's rarely necessary, but at the other end of the spectrum there are classes where you couldn't get a good grade without it.

Getting a good grade in a class on x is so different from learning a lot about x that you have to choose one or the other, and you can't blame students if they choose grades. Everyone judges them by their grades — graduate programs, employers, scholarships, even their own parents.

I liked learning, and I really enjoyed some of the papers and programs I wrote in college. But did I ever, after turning in a paper in some class, sit down and write another just for fun? Of course not. I had things due in other classes. If it ever came to a choice of learning or grades, I chose grades. I hadn't come to college to do badly.

Anyone who cares about getting good grades has to play this game, or they'll be surpassed by those who do. And at elite universities, that means nearly everyone, since someone who didn't care about getting good grades probably wouldn't be there in the first place. The result is that students compete to maximize the difference between learning and getting good grades.

Why are tests so bad? More precisely, why are they so hackable? Any experienced programmer could answer that. How hackable is software whose author hasn't paid any attention to preventing it from being hacked? Usually it's as porous as a colander.

Hackable is the default for any test imposed by an authority. The reason the

tests you're given are so consistently bad — so consistently far from measuring what they're supposed to measure — is simply that the people creating them haven't made much effort to prevent them from being hacked.

But you can't blame teachers if their tests are hackable. Their job is to teach, not to create unhackable tests. The real problem is grades, or more precisely, that grades have been overloaded. If grades were merely a way for teachers to tell students what they were doing right and wrong, like a coach giving advice to an athlete, students wouldn't be tempted to hack tests. But unfortunately after a certain age grades become more than advice. After a certain age, whenever you're being taught, you're usually also being judged.

I've used college tests as an example, but those are actually the least hackable. All the tests most students take their whole lives are at least as bad, including, most spectacularly of all, the test that gets them into college. If getting into college were merely a matter of having the quality of one's mind measured by admissions officers the way scientists measure the mass of an object, we could tell teenage kids "learn a lot" and leave it at that. You can tell how bad college admissions are, as a test, from how unlike high school that sounds. In practice, the freakishly specific nature of the stuff ambitious kids have to do in high school is directly proportionate to the hackability of college admissions. The classes you don't care about that are mostly memorization, the random "extracurricular activities" you have to participate in to show you're "well-rounded," the standardized tests as artificial as chess, the "essay" you have to write that's presumably meant to hit some very specific target, but you're not told what.

As well as being bad in what it does to kids, this test is also bad in the sense of being very hackable. So hackable that whole industries have grown up to hack it. This is the explicit purpose of test-prep companies and admissions counsellors, but it's also a significant part of the function of private schools.

Why is this particular test so hackable? I think because of what it's measuring. Although the popular story is that the way to get into a good college is to be really smart, admissions officers at elite colleges neither are, nor claim to be, looking only for that. What are they looking for? They're looking for people who are not simply smart, but admirable in some more general sense. And how is this more general admirableness measured? The admissions officers feel it. In other words, they accept who they like.

So what college admissions is a test of is whether you suit the taste of some group of people. Well, of course a test like that is going to be hackable. And because it's both very hackable and there's (thought to be) a lot at stake, it's hacked like nothing else. That's why it distorts your life so much for so long.

It's no wonder high school students often feel alienated. The shape of their lives is completely artificial.

But wasting your time is not the worst thing the educational system does to you. The worst thing it does is to train you that the way to win is by hacking

bad tests. This is a much subtler problem that I didn't recognize until I saw it happening to other people.

When I started advising startup founders at Y Combinator, especially young ones, I was puzzled by the way they always seemed to make things overcomplicated. How, they would ask, do you raise money? What's the trick for making venture capitalists want to invest in you? The best way to make VCs want to invest in you, I would explain, is to actually be a good investment. Even if you could trick VCs into investing in a bad startup, you'd be tricking yourselves too. You're investing time in the same company you're asking them to invest money in. If it's not a good investment, why are you even doing it?

Oh, they'd say, and then after a pause to digest this revelation, they'd ask: What makes a startup a good investment?

So I would explain that what makes a startup promising, not just in the eyes of investors but in fact, is *growth*. Ideally in revenue, but failing that in usage. What they needed to do was get lots of users.

How does one get lots of users? They had all kinds of ideas about that. They needed to do a big launch that would get them "exposure." They needed influential people to talk about them. They even knew they needed to launch on a tuesday, because that's when one gets the most attention.

No, I would explain, that is not how to get lots of users. The way you get lots of users is to make the product really great. Then people will not only use it but recommend it to their friends, so your growth will be exponential once you *get it started*.

At this point I've told the founders something you'd think would be completely obvious: that they should make a good company by making a good product. And yet their reaction would be something like the reaction many physicists must have had when they first heard about the theory of relativity: a mixture of astonishment at its apparent genius, combined with a suspicion that anything so weird couldn't possibly be right. Ok, they would say, dutifully. And could you introduce us to such-and-such influential person? And remember, we want to launch on Tuesday.

It would sometimes take founders years to grasp these simple lessons. And not because they were lazy or stupid. They just seemed blind to what was right in front of them.

Why, I would ask myself, do they always make things so complicated? And then one day I realized this was not a rhetorical question.

Why did founders tie themselves in knots doing the wrong things when the answer was right in front of them? Because that was what they'd been trained to do. Their education had taught them that the way to win was to hack the test. And without even telling them they were being trained to do this. The younger ones, the recent graduates, had never faced a non-artificial test. They

thought this was just how the world worked: that the first thing you did, when facing any kind of challenge, was to figure out what the trick was for hacking the test. That's why the conversation would always start with how to raise money, because that read as the test. It came at the end of YC. It had numbers attached to it, and higher numbers seemed to be better. It must be the test.

There are certainly big chunks of the world where the way to win is to hack the test. This phenomenon isn't limited to schools. And some people, either due to ideology or ignorance, claim that this is true of startups too. But it isn't. In fact, one of the most striking things about startups is the degree to which you win by simply doing good work. There are edge cases, as there are in anything, but in general you win by getting users, and what users care about is whether the product does what they want.

Why did it take me so long to understand why founders made startups overcomplicated? Because I hadn't realized explicitly that schools train us to win by hacking bad tests. And not just them, but me! I'd been trained to hack bad tests too, and hadn't realized it till decades later.

I had lived as if I realized it, but without knowing why. For example, I had avoided working for big companies. But if you'd asked why, I'd have said it was because they were bogus, or bureaucratic. Or just yuck. I never understood how much of my dislike of big companies was due to the fact that you win by hacking bad tests.

Similarly, the fact that the tests were unhackable was a lot of what attracted me to startups. But again, I hadn't realized that explicitly.

I had in effect achieved by successive approximations something that may have a closed-form solution. I had gradually undone my training in hacking bad tests without knowing I was doing it. Could someone coming out of school banish this demon just by knowing its name, and saying begone? It seems worth trying.

Merely talking explicitly about this phenomenon is likely to make things better, because much of its power comes from the fact that we take it for granted. After you've noticed it, it seems the elephant in the room, but it's a pretty well camouflaged elephant. The phenomenon is so old, and so pervasive. And it's simply the result of neglect. No one meant things to be this way. This is just what happens when you combine learning with grades, competition, and the naive assumption of unhackability.

It was mind-blowing to realize that two of the things I'd puzzled about the most — the bogusness of high school, and the difficulty of getting founders to see the obvious — both had the same cause. It's rare for such a big block to slide into place so late.

Usually when that happens it has implications in a lot of different areas, and this case seems no exception. For example, it suggests both that education could be done better, and how you might fix it. But it also suggests a potential answer to the question all big companies seem to have: how can we be more like a startup?

I'm not going to chase down all the implications now. What I want to focus on here is what it means for individuals.

To start with, it means that most ambitious kids graduating from college have something they may want to unlearn. But it also changes how you look at the world. Instead of looking at all the different kinds of work people do and thinking of them vaguely as more or less appealing, you can now ask a very specific question that will sort them in an interesting way: to what extent do you win at this kind of work by hacking bad tests?

It would help if there was a way to recognize bad tests quickly. Is there a pattern here? It turns out there is.

Tests can be divided into two kinds: those that are imposed by authorities, and those that aren't. Tests that aren't imposed by authorities are inherently unhackable, in the sense that no one is claiming they're tests of anything more than they actually test. A football match, for example, is simply a test of who wins, not which team is better. You can tell that from the fact that commentators sometimes say afterward that the better team won. Whereas tests imposed by authorities are usually proxies for something else. A test in a class is supposed to measure not just how well you did on that particular test, but how much you learned in the class. While tests that aren't imposed by authorities are inherently unhackable, those imposed by authorities have to be made unhackable. Usually they aren't. So as a first approximation, bad tests are roughly equivalent to tests imposed by authorities.

You might actually like to win by hacking bad tests. Presumably some people do. But I bet most people who find themselves doing this kind of work don't like it. They just take it for granted that this is how the world works, unless you want to drop out and be some kind of hippie artisan.

I suspect many people implicitly assume that working in a field with bad tests is the price of making lots of money. But that, I can tell you, is false. It used to be true. In the mid-twentieth century, when the economy was *composed of oligopolies*, the only way to the top was by playing their game. But it's not true now. There are now ways to get rich by doing good work, and that's part of the reason people are so much more excited about getting rich than they used to be. When I was a kid, you could either become an engineer and make cool things, or make lots of money by becoming an "executive." Now you can make lots of money by making cool things.

Hacking bad tests is becoming less important as the link between work and authority erodes. The erosion of that link is one of the most important trends happening now, and we see its effects in almost every kind of work people do. Startups are one of the most visible examples, but we see much the same thing in writing. Writers no longer have to submit to publishers and editors to reach readers; now they can go direct.

The more I think about this question, the more optimistic I get. This seems one

of those situations where we don't realize how much something was holding us back until it's eliminated. And I can foresee the whole bogus edifice crumbling. Imagine what happens as more and more people start to ask themselves if they want to win by hacking bad tests, and decide that they don't. The kinds of work where you win by hacking bad tests will be starved of talent, and the kinds where you win by doing good work will see an influx of the most ambitious people. And as hacking bad tests shrinks in importance, education will evolve to stop training us to do it. Imagine what the world could look like if that happened.

This is not just a lesson for individuals to unlearn, but one for society to unlearn, and we'll be amazed at the energy that's liberated when we do.

Notes

1 If using tests only to measure learning sounds impossibly utopian, that is already the way things work at Lambda School. Lambda School doesn't have grades. You either graduate or you don't. The only purpose of tests is to decide at each stage of the curriculum whether you can continue to the next. So in effect the whole school is pass/fail.

2 If the final exam consisted of a long conversation with the professor, you could prepare for it by reading good books on medieval history. A lot of the hackability of tests in schools is due to the fact that the same test has to be given to large numbers of students.

3 Learning is the naive algorithm for getting good grades.

4 *Hacking* has multiple senses. There's a narrow sense in which it means to compromise something. That's the sense in which one hacks a bad test. But there's another, more general sense, meaning to find a surprising solution to a problem, often by thinking differently about it. Hacking in this sense is a wonderful thing. And indeed, some of the hacks people use on bad tests are impressively ingenious; the problem is not so much the hacking as that, because the tests are hackable, they don't test what they're meant to.

5 The people who pick startups at Y Combinator are similar to admissions officers, except that instead of being arbitrary, their acceptance criteria are trained by a very tight feedback loop. If you accept a bad startup or reject a good one, you will usually know it within a year or two at the latest, and often within a month.

6 I'm sure admissions officers are tired of reading applications from kids who seem to have no personality beyond being willing to seem however they're supposed to seem to get accepted. What they don't realize is that they are, in a sense, looking in a mirror. The lack of authenticity in the applicants is a reflection of the arbitrariness of the application process. A dictator might just as well complain about the lack of authenticity in the people around him.

7 By good work, I don't mean morally good, but good in the sense in which a good craftsman does good work.

8 There are borderline cases where it's hard to say which category a test falls in. For example, is raising venture capital like college admissions, or is it like selling to a customer?

9 Note that a good test is merely one that's unhackable. Good here doesn't mean morally good, but good in the sense of working well. The difference between fields with bad tests and good ones is not that the former are bad and the latter are good, but that the former are bogus and the latter aren't. But those two measures are not unrelated. As Tara Ploughman said, the path from good to evil goes through bogus.

10 People who think the recent increase in *economic inequality* is due to changes in tax policy seem very naive to anyone with experience in startups. Different people are getting rich now than used to, and they're getting much richer than mere tax savings could make them.

11 Note to tiger parents: you may think you're training your kids to win, but if you're training them to win by hacking bad tests, you are, as parents so often do, training them to fight the last war.

Thanks to Austen Allred, Trevor Blackwell, Patrick Collison, Jessica Livingston, Robert Morris, and Harj Taggar for reading drafts of this.

Russian Translation Arabic Translation

Swedish Translation

196 Billionaires Build

December 2020

As I was deciding what to write about next, I was surprised to find that two separate essays I'd been planning to write were actually the same.

The first is about how to ace your Y Combinator interview. There has been so much nonsense written about this topic that I've been meaning for years to write something telling founders the truth.

The second is about something politicians sometimes say — that the only way to become a billionaire is by exploiting people — and why this is mistaken.

Keep reading, and you'll learn both simultaneously.

I know the politicians are mistaken because it was my job to predict which people will become billionaires. I think I can truthfully say that I know as much about how to do this as anyone. If the key to becoming a billionaire — the defining feature of billionaires — was to exploit people, then I, as a professional billionaire scout, would surely realize this and look for people who would be good at it, just as an NFL scout looks for speed in wide receivers.

But aptitude for exploiting people is not what Y Combinator looks for at all. In fact, it's the opposite of what they look for. I'll tell you what they do look for, by explaining how to convince Y Combinator to fund you, and you can see for yourself.

What YC looks for, above all, is founders who understand some group of users and can make what they want. This is so important that it's YC's motto: "Make something people want."

A big company can to some extent force unsuitable products on unwilling customers, but a startup doesn't have the power to do that. A startup must sing for its supper, by making things that genuinely delight its customers. Otherwise it will never get off the ground.

Here's where things get difficult, both for you as a founder and for the YC partners trying to decide whether to fund you. In a market economy, it's hard to make something people want that they don't already have. That's the great thing about market economies. If other people both knew about this need and were able to satisfy it, they already would be, and there would be no room for your startup.

Which means the conversation during your YC interview will have to be about something new: either a new need, or a new way to satisfy one. And not just new, but uncertain. If it were certain that the need existed and that you could satisfy it, that certainty would be reflected in large and rapidly growing revenues, and you wouldn't be seeking seed funding.

So the YC partners have to guess both whether you've discovered a real need, and whether you'll be able to satisfy it. That's what they are, at least in this part of their job: professional guessers. They have 1001 heuristics for doing this, and I'm not going to tell you all of them, but I'm happy to tell you the most important ones, because these can't be faked; the only way to "hack" them would be to do what you should be doing anyway as a founder.

The first thing the partners will try to figure out, usually, is whether what you're making will ever be something a lot of people want. It doesn't have to be something a lot of people want now. The product and the market will both evolve, and will influence each other's evolution. But in the end there has to be something with a huge market. That's what the partners will be trying to figure out: is there a path to a huge market? 1

Sometimes it's obvious there will be a huge market. If *Boom* manages to ship an airliner at all, international airlines will have to buy it. But usually it's not obvious. Usually the path to a huge market is by growing a small market. This idea is important enough that it's worth coining a phrase for, so let's call one of these small but growable markets a "larval market."

The perfect example of a larval market might be Apple's market when they were founded in 1976. In 1976, not many people wanted their own computer. But

more and more started to want one, till now every 10 year old on the planet wants a computer (but calls it a "phone").

The ideal combination is the group of founders who are *"living in the future"* in the sense of being at the leading edge of some kind of change, and who are building something they themselves want. Most super-successful startups are of this type. Steve Wozniak wanted a computer. Mark Zuckerberg wanted to engage online with his college friends. Larry and Sergey wanted to find things on the web. All these founders were building things they and their peers wanted, and the fact that they were at the leading edge of change meant that more people would want these things in the future.

But although the ideal larval market is oneself and one's peers, that's not the only kind. A larval market might also be regional, for example. You build something to serve one location, and then expand to others.

The crucial feature of the initial market is that it exist. That may seem like an obvious point, but the lack of it is the biggest flaw in most startup ideas. There have to be some people who want what you're building right now, and want it so urgently that they're willing to use it, bugs and all, even though you're a small company they've never heard of. There don't have to be many, but there have to be some. As long as you have some users, there are straightforward ways to get more: build new features they want, seek out more people like them, get them to refer you to their friends, and so on. But these techniques all require some initial seed group of users.

So this is one thing the YC partners will almost certainly dig into during your interview. Who are your first users going to be, and how do you know they want this? If I had to decide whether to fund startups based on a single question, it would be "How do you know people want this?"

The most convincing answer is "Because we and our friends want it." It's even better when this is followed by the news that you've already built a prototype, and even though it's very crude, your friends are using it, and it's spreading by word of mouth. If you can say that and you're not lying, the partners will switch from default no to default yes. Meaning you're in unless there's some other disqualifying flaw.

That is a hard standard to meet, though. Airbnb didn't meet it. They had the first part. They had made something they themselves wanted. But it wasn't spreading. So don't feel bad if you don't hit this gold standard of convincingness. If Airbnb didn't hit it, it must be too high.

In practice, the YC partners will be satisfied if they feel that you have a deep understanding of your users' needs. And the Airbnbs did have that. They were able to tell us all about what motivated hosts and guests. They knew from first-hand experience, because they'd been the first hosts. We couldn't ask them a question they didn't know the answer to. We ourselves were not very excited about the idea as users, but we knew this didn't prove anything, because there

were lots of successful startups we hadn't been excited about as users. We were able to say to ourselves "They seem to know what they're talking about. Maybe they're onto something. It's not growing yet, but maybe they can figure out how to make it grow during YC." Which they did, about three weeks into the batch.

The best thing you can do in a YC interview is to teach the partners about your users. So if you want to prepare for your interview, one of the best ways to do it is to go talk to your users and find out exactly what they're thinking. Which is what you should be doing anyway.

This may sound strangely credulous, but the YC partners want to rely on the founders to tell them about the market. Think about how VCs typically judge the potential market for an idea. They're not ordinarily domain experts themselves, so they forward the idea to someone who is, and ask for their opinion. YC doesn't have time to do this, but if the YC partners can convince themselves that the founders both (a) know what they're talking about and (b) aren't lying, they don't need outside domain experts. They can use the founders themselves as domain experts when evaluating their own idea.

This is why YC interviews aren't pitches. To give as many founders as possible a chance to get funded, we made interviews as short as we could: 10 minutes. That is not enough time for the partners to figure out, through the indirect evidence in a pitch, whether you know what you're talking about and aren't lying. They need to dig in and ask you questions. There's not enough time for sequential access. They need random access. 2

The worst advice I ever heard about how to succeed in a YC interview is that you should take control of the interview and make sure to deliver the message you want to. In other words, turn the interview into a pitch. (elaborate expletive). It is so annoying when people try to do that. You ask them a question, and instead of answering it, they deliver some obviously prefabricated blob of pitch. It eats up 10 minutes really fast.

There is no one who can give you accurate advice about what to do in a YC interview except a current or former YC partner. People who've merely been interviewed, even successfully, have no idea of this, but interviews take all sorts of different forms depending on what the partners want to know about most. Sometimes they're all about the founders, other times they're all about the idea. Sometimes some very narrow aspect of the idea. Founders sometimes walk away from interviews complaining that they didn't get to explain their idea completely. True, but they explained enough.

Since a YC interview consists of questions, the way to do it well is to answer them well. Part of that is answering them candidly. The partners don't expect you to know everything. But if you don't know the answer to a question, don't try to bullshit your way out of it. The partners, like most experienced investors, are professional bullshit detectors, and you are (hopefully) an amateur bullshitter. And if you try to bullshit them and fail, they may not even tell you that you failed. So it's better to be honest than to try to sell them. If you don't know the

answer to a question, say you don't, and tell them how you'd go about finding it, or tell them the answer to some related question.

If you're asked, for example, what could go wrong, the worst possible answer is "nothing." Instead of convincing them that your idea is bullet-proof, this will convince them that you're a fool or a liar. Far better to go into gruesome detail. That's what experts do when you ask what could go wrong. The partners know that your idea is risky. That's what a good bet looks like at this stage: a tiny probability of a huge outcome.

Ditto if they ask about competitors. Competitors are rarely what kills startups. Poor execution does. But you should know who your competitors are, and tell the YC partners candidly what your relative strengths and weaknesses are. Because the YC partners know that competitors don't kill startups, they won't hold competitors against you too much. They will, however, hold it against you if you seem either to be unaware of competitors, or to be minimizing the threat they pose. They may not be sure whether you're clueless or lying, but they don't need to be.

The partners don't expect your idea to be perfect. This is seed investing. At this stage, all they can expect are promising hypotheses. But they do expect you to be thoughtful and honest. So if trying to make your idea seem perfect causes you to come off as glib or clueless, you've sacrificed something you needed for something you didn't.

If the partners are sufficiently convinced that there's a path to a big market, the next question is whether you'll be able to find it. That in turn depends on three things: the general qualities of the founders, their specific expertise in this domain, and the relationship between them. How determined are the founders? Are they good at building things? Are they resilient enough to keep going when things go wrong? How strong is their friendship?

Though the Airbnbs only did ok in the idea department, they did spectacularly well in this department. The story of how they'd funded themselves by making Obama- and McCain-themed breakfast cereal was the single most important factor in our decision to fund them. They didn't realize it at the time, but what seemed to them an irrelevant story was in fact fabulously good evidence of their qualities as founders. It showed they were resourceful and determined, and could work together.

It wasn't just the cereal story that showed that, though. The whole interview showed that they cared. They weren't doing this just for the money, or because startups were cool. The reason they were working so hard on this company was because it was their project. They had discovered an interesting new idea, and they just couldn't let it go.

Mundane as it sounds, that's the most powerful motivator of all, not just in startups, but in most ambitious undertakings: to be *genuinely interested* in what you're building. This is what really drives billionaires, or at least the ones who

become billionaires from starting companies. The company is their project.

One thing few people realize about billionaires is that all of them could have stopped sooner. They could have gotten acquired, or found someone else to run the company. Many founders do. The ones who become really rich are the ones who keep working. And what makes them keep working is not just money. What keeps them working is the same thing that keeps anyone else working when they could stop if they wanted to: that there's nothing else they'd rather do.

That, not exploiting people, is the defining quality of people who become billionaires from starting companies. So that's what YC looks for in founders: authenticity. People's motives for starting startups are usually mixed. They're usually doing it from some combination of the desire to make money, the desire to seem cool, genuine interest in the problem, and unwillingness to work for someone else. The last two are more powerful motivators than the first two. It's ok for founders to want to make money or to seem cool. Most do. But if the founders seem like they're doing it *just* to make money or *just* to seem cool, they're not likely to succeed on a big scale. The founders who are doing it for the money will take the first sufficiently large acquisition offer, and the ones who are doing it to seem cool will rapidly discover that there are much less painful ways of seeming cool. 3

Y Combinator certainly sees founders whose m.o. is to exploit people. YC is a magnet for them, because they want the YC brand. But when the YC partners detect someone like that, they reject them. If bad people made good founders, the YC partners would face a moral dilemma. Fortunately they don't, because bad people make bad founders. This exploitative type of founder is not going to succeed on a large scale, and in fact probably won't even succeed on a small one, because they're always going to be taking shortcuts. They see YC itself as a shortcut.

Their exploitation usually begins with their own cofounders, which is disastrous, since the cofounders' relationship is the foundation of the company. Then it moves on to the users, which is also disastrous, because the sort of early adopters a successful startup wants as its initial users are the hardest to fool. The best this kind of founder can hope for is to keep the edifice of deception tottering along until some acquirer can be tricked into buying it. But that kind of acquisition is never very big. 4

If professional billionaire scouts know that exploiting people is not the skill to look for, why do some politicians think this is the defining quality of billionaires?

I think they start from the feeling that it's wrong that one person could have so much more money than another. It's understandable where that feeling comes from. It's in our DNA, and even in the DNA of other species.

If they limited themselves to saying that it made them feel bad when one person had so much more money than other people, who would disagree? It makes me feel bad too, and I think people who make a lot of money have a moral obligation

to use it for the common good. The mistake they make is to jump from feeling bad that some people are much richer than others to the conclusion that there's no legitimate way to make a very large amount of money. Now we're getting into statements that are not only falsifiable, but false.

There are certainly some people who become rich by doing bad things. But there are also plenty of people who behave badly and don't make that much from it. There is no correlation — in fact, probably an inverse correlation — between how badly you behave and how much money you make.

The greatest danger of this nonsense may not even be that it sends policy astray, but that it misleads ambitious people. Can you imagine a better way to destroy social mobility than by telling poor kids that the way to get rich is by exploiting people, while the rich kids know, from having watched the preceding generation do it, how it's really done?

I'll tell you how it's really done, so you can at least tell your own kids the truth. It's all about users. The most reliable way to become a billionaire is to start a company that *grows fast*, and the way to grow fast is to make what users want. Newly started startups have no choice but to delight users, or they'll never even get rolling. But this never stops being the lodestar, and bigger companies take their eye off it at their peril. Stop delighting users, and eventually someone else will.

Users are what the partners want to know about in YC interviews, and what I want to know about when I talk to founders that we funded ten years ago and who are billionaires now. What do users want? What new things could you build for them? Founders who've become billionaires are always eager to talk about that topic. That's how they became billionaires.

Notes

[1] The YC partners have so much practice doing this that they sometimes see paths that the founders themselves haven't seen yet. The partners don't try to seem skeptical, as buyers in transactions often do to increase their leverage. Although the founders feel their job is to convince the partners of the potential of their idea, these roles are not infrequently reversed, and the founders leave the interview feeling their idea has more potential than they realized.

[2] In practice, 7 minutes would be enough. You rarely change your mind at minute 8. But 10 minutes is socially convenient.

[3] I myself took the first sufficiently large acquisition offer in my first startup, so I don't blame founders for doing this. There's nothing wrong with starting a startup to make money. You need to make money somehow, and for some people startups are the most efficient way to do it. I'm just saying that these are not the startups that get really big.

[4] Not these days, anyway. There were some big ones during the Internet Bubble, and indeed some big IPOs.

Thanks to Trevor Blackwell, Jessica Livingston, Robert Morris, Geoff Ralston, and Harj Taggar for reading drafts of this.

197 Earnestness

December 2020

Jessica and I have certain words that have special significance when we're talking about startups. The highest compliment we can pay to founders is to describe them as "earnest." This is not by itself a guarantee of success. You could be earnest but incapable. But when founders are both formidable (another of our words) and earnest, they're as close to unstoppable as you get.

Earnestness sounds like a boring, even Victorian virtue. It seems a bit of an anachronism that people in Silicon Valley would care about it. Why does this matter so much?

When you call someone earnest, you're making a statement about their motives. It means both that they're doing something for the right reasons, and that they're trying as hard as they can. If we imagine motives as vectors, it means both the direction and the magnitude are right. Though these are of course related: when people are doing something for the right reasons, they try harder.

1

The reason motives matter so much in Silicon Valley is that so many people there have the wrong ones. Starting a successful startup makes you rich and famous. So a lot of the people trying to start them are doing it for those reasons. Instead of what? Instead of interest in the problem for its own sake. That is the root of earnestness. 2

It's also the hallmark of a nerd. Indeed, when people describe themselves as "x nerds," what they mean is that they're interested in x for its own sake, and not because it's cool to be interested in x, or because of what they can get from it. They're saying they care so much about x that they're willing to sacrifice seeming cool for its sake.

A *genuine interest* in something is a very powerful motivator — for some people, the most powerful motivator of all. 3 Which is why it's what Jessica and I look for in founders. But as well as being a source of strength, it's also a source of vulnerability. Caring constrains you. The earnest can't easily reply in kind to mocking banter, or put on a cool facade of nihil admirari. They care too much. They are doomed to be the straight man. That's a real disadvantage in your *teenage years*, when mocking banter and nihil admirari often have the upper hand. But it becomes an advantage later.

It's a commonplace now that the kids who were nerds in high school become the cool kids' bosses later on. But people misunderstand why this happens. It's

not just because the nerds are smarter, but also because they're more earnest. When the problems get harder than the fake ones you're given in high school, caring about them starts to matter.

Does it always matter? Do the earnest always win? Not always. It probably doesn't matter much in politics, or in crime, or in certain types of business that are similar to crime, like gambling, personal injury law, patent trolling, and so on. Nor does it matter in academic fields at the more *bogus* end of the spectrum. And though I don't know enough to say for sure, it may not matter in some kinds of humor: it may be possible to be completely cynical and still be very funny. 4

Looking at the list of fields I mentioned, there's an obvious pattern. Except possibly for humor, these are all types of work I'd avoid like the plague. So that could be a useful heuristic for deciding which fields to work in: how much does earnestness matter? Which can in turn presumably be inferred from the prevalence of nerds at the top.

Along with "nerd," another word that tends to be associated with earnestness is "naive." The earnest often seem naive. It's not just that they don't have the motives other people have. They often don't fully grasp that such motives exist. Or they may know intellectually that they do, but because they don't feel them, they forget about them. 5

It works to be slightly naive not just about motives but also, believe it or not, about the problems you're working on. Naive optimism can compensate for the bit rot that *rapid change* causes in established beliefs. You plunge into some problem saying "How hard can it be?", and then after solving it you learn that it was till recently insoluble.

Naivete is an obstacle for anyone who wants to seem sophisticated, and this is one reason would-be intellectuals find it so difficult to understand Silicon Valley. It hasn't been safe for such people to use the word "earnest" outside scare quotes since Oscar Wilde wrote "The Importance of Being Earnest" in 1895. And yet when you zoom in on Silicon Valley, right into *Jessica Livingston's brain*, that's what her x-ray vision is seeking out in founders. Earnestness! Who'd have guessed? Reporters literally can't believe it when founders making piles of money say that they started their companies to make the world better. The situation seems made for mockery. How can these founders be so naive as not to realize how implausible they sound?

Though those asking this question don't realize it, that's not a rhetorical question.

A lot of founders are faking it, of course, particularly the smaller fry, and the soon to be smaller fry. But not all of them. There are a significant number of founders who really are interested in the problem they're solving mainly for its own sake.

Why shouldn't there be? We have no difficulty believing that people would be interested in history or math or even old bus tickets for their own sake. Why

can't there be people interested in self-driving cars or social networks for their own sake? When you look at the question from this side, it seems obvious there would be. And isn't it likely that having a deep interest in something would be a source of great energy and resilience? It is in every other field.

The question really is why we have a blind spot about business. And the answer to that is obvious if you know enough history. For most of history, making large amounts of money has not been very intellectually interesting. In preindustrial times it was never far from robbery, and some areas of business still retain that character, except using lawyers instead of soldiers.

But there are other areas of business where the work is genuinely interesting. Henry Ford got to spend much of his time working on interesting technical problems, and for the last several decades the trend in that direction has been accelerating. It's much easier now to make a lot of money by working on something you're interested in than it was *50 years ago*. And that, rather than how fast they grow, may be the most important change that startups represent. Though indeed, the fact that the work is genuinely interesting is a big part of why it gets done so fast. 6

Can you imagine a more important change than one in the relationship between intellectual curiosity and money? These are two of the most powerful forces in the world, and in my lifetime they've become significantly more aligned. How could you not be fascinated to watch something like this happening in real time?

I meant this essay to be about earnestness generally, and now I've gone and talked about startups again. But I suppose at least it serves as an example of an x nerd in the wild.

Notes

[1] It's interesting how many different ways there are *not* to be earnest: to be cleverly cynical, to be superficially brilliant, to be conspicuously virtuous, to be cool, to be sophisticated, to be orthodox, to be a snob, to bully, to pander, to be on the make. This pattern suggests that earnestness is not one end of a continuum, but a target one can fall short of in multiple dimensions.

Another thing I notice about this list is that it sounds like a list of the ways people behave on Twitter. Whatever else social media is, it's a vivid catalogue of ways not to be earnest.

[2] People's motives are as mixed in Silicon Valley as anywhere else. Even the founders motivated mostly by money tend to be at least somewhat interested in the problem they're solving, and even the founders most interested in the problem they're solving also like the idea of getting rich. But there's great variation in the relative proportions of different founders' motivations.

And when I talk about "wrong" motives, I don't mean morally wrong. There's nothing morally wrong with starting a startup to make money. I just mean that those startups don't do as well.

[3] The most powerful motivator for most people is probably family. But there are some for whom intellectual curiosity comes first. In his (wonderful) autobiography, Paul Halmos says explicitly that for a mathematician, math must come before anything else, including family. Which at least implies that it did for him.

[4] Interestingly, just as the word "nerd" implies earnestness even when used as a metaphor, the word "politics" implies the opposite. It's not only in actual politics that earnestness seems to be a handicap, but also in office politics and academic politics.

[5] It's a bigger social error to seem naive in most European countries than it is in America, and this may be one of subtler reasons startups are less common there. Founder culture is completely at odds with sophisticated cynicism.

The most earnest part of Europe is Scandinavia, and not surprisingly this is also the region with the highest number of successful startups per capita.

[6] Much of business is schleps, and probably always will be. But even being a professor is largely schleps. It would be interesting to collect statistics about the schlep ratios of different jobs, but I suspect they'd rarely be less than 30%.

Thanks to Trevor Blackwell, Patrick Collison, Suhail Doshi, Jessica Livingston, Mattias Ljungman, Harj Taggar, and Kyle Vogt for reading drafts of this.

200 Write Simply

March 2021

I try to write using ordinary words and simple sentences.

That kind of writing is easier to read, and the easier something is to read, the more deeply readers will engage with it. The less energy they expend on your prose, the more they'll have left for your ideas.

And the further they'll read. Most readers' energy tends to flag part way through an article or essay. If the friction of reading is low enough, more keep going till the end.

There's an Italian dish called *saltimbocca*, which means "leap into the mouth." My goal when writing might be called *saltintesta*: the ideas leap into your head and you barely notice the words that got them there.

It's too much to hope that writing could ever be pure ideas. You might not even want it to be. But for most writers, most of the time, that's the goal to aim for. The gap between most writing and pure ideas is not filled with poetry.

Plus it's more considerate to write simply. When you write in a fancy way to impress people, you're making them do extra work just so you can seem cool.

It's like trailing a long train behind you that readers have to carry.

And remember, if you're writing in English, that a lot of your readers won't be native English speakers. Their understanding of ideas may be way ahead of their understanding of English. So you can't assume that writing about a difficult topic means you can use difficult words.

Of course, fancy writing doesn't just conceal ideas. It can also conceal the lack of them. That's why some people write that way, to conceal the fact that they have ___nothing to say. Whereas writing simply keeps you honest. If you say nothing simply, it will be obvious to everyone, including you.

Simple writing also lasts better. People reading your stuff in the future will be in much the same position as people from other countries reading it today. The culture and the language will have changed. It's not vain to care about that, any more than it's vain for a woodworker to build a chair to last.

Indeed, lasting is not merely an accidental quality of chairs, or writing. It's a sign you did a good job.

But although these are all real advantages of writing simply, none of them are why I do it. The main reason I write simply is that it offends me not to. When I write a sentence that seems too complicated, or that uses unnecessarily intellectual words, it doesn't seem fancy to me. It seems clumsy.

There are of course times when you want to use a complicated sentence or fancy word for effect. But you should never do it by accident.

The other reason my writing ends up being simple is the way I do it. I write the first draft fast, then spend days editing it, trying to get everything just right. Much of this editing is cutting, and that makes simple writing even simpler.

207 How to Work Hard

June 2021

It might not seem there's much to learn about how to work hard. Anyone who's been to school knows what it entails, even if they chose not to do it. There are 12 year olds who work amazingly hard. And yet when I ask if I know more about working hard now than when I was in school, the answer is definitely yes.

One thing I know is that if you want to do great things, you'll have to work very hard. I wasn't sure of that as a kid. Schoolwork varied in difficulty; one didn't always have to work super hard to do well. And some of the things famous adults did, they seemed to do almost effortlessly. Was there, perhaps, some way to evade hard work through sheer brilliance? Now I know the answer to that question. There isn't.

The reason some subjects seemed easy was that my school had low standards. And the reason famous adults seemed to do things effortlessly was years of practice; they made it look easy.

Of course, those famous adults usually had a lot of natural ability too. There are three ingredients in great work: natural ability, practice, and effort. You can do pretty well with just two, but to do the best work you need all three: you need great natural ability *and* to have practiced a lot *and* to be trying very hard. 1

Bill Gates, for example, was among the smartest people in business in his era, but he was also among the hardest working. "I never took a day off in my twenties," he said. "Not one." It was similar with Lionel Messi. He had great natural ability, but when his youth coaches talk about him, what they remember is not his talent but his dedication and his desire to win. P. G. Wodehouse would probably get my vote for best English writer of the 20th century, if I had to choose. Certainly no one ever made it look easier. But no one ever worked harder. At 74, he wrote

with each new book of mine I have, as I say, the feeling that this time I > have picked a lemon in the garden of literature. A good thing, really, I > suppose. Keeps one up on one's toes and makes one rewrite every sentence ten > times. Or in many cases twenty times.

Sounds a bit extreme, you think. And yet Bill Gates sounds even more extreme. Not one day off in ten years? These two had about as much natural ability as anyone could have, and yet they also worked about as hard as anyone could work. You need both.

That seems so obvious, and yet in practice we find it slightly hard to grasp. There's a faint xor between talent and hard work. It comes partly from popular culture, where it seems to run very deep, and partly from the fact that the outliers are so rare. If great talent and great drive are both rare, then people with both are rare squared. Most people you meet who have a lot of one will have less of the other. But you'll need both if you want to be an outlier yourself. And since you can't really change how much natural talent you have, in practice doing great work, insofar as you can, reduces to working very hard.

It's straightforward to work hard if you have clearly defined, externally imposed goals, as you do in school. There is some technique to it: you have to learn not to lie to yourself, not to procrastinate (which is a form of lying to yourself), not to get distracted, and not to give up when things go wrong. But this level of discipline seems to be within the reach of quite young children, if they want it.

What I've learned since I was a kid is how to work toward goals that are neither clearly defined nor externally imposed. You'll probably have to learn both if you want to do really great things.

The most basic level of which is simply to feel you should be working without anyone telling you to. Now, when I'm not working hard, alarm bells go off. I

can't be sure I'm getting anywhere when I'm working hard, but I can be sure I'm getting nowhere when I'm not, and it feels awful. 2

There wasn't a single point when I learned this. Like most little kids, I enjoyed the feeling of achievement when I learned or did something new. As I grew older, this morphed into a feeling of disgust when I wasn't achieving anything. The one precisely dateable landmark I have is when I stopped watching TV, at age 13.

Several people I've talked to remember getting serious about work around this age. When I asked Patrick Collison when he started to find idleness distasteful, he said

I think around age 13 or 14. I have a clear memory from around then of > sitting in the sitting room, staring outside, and wondering why I was > wasting my summer holiday.

Perhaps something changes at adolescence. That would make sense.

Strangely enough, the biggest obstacle to getting serious about work was probably school, which made work (what they called work) seem boring and pointless. I had to learn what real work was before I could wholeheartedly desire to do it. That took a while, because even in college a lot of the work is pointless; there are entire departments that are pointless. But as I learned the shape of real work, I found that my desire to do it slotted into it as if they'd been made for each other.

I suspect most people have to learn what work is before they can love it. Hardy wrote eloquently about this in *A Mathematician's Apology* :

I do not remember having felt, as a boy, any *passion* for mathematics, and > such notions as I may have had of the career of a mathematician were far > from noble. I thought of mathematics in terms of examinations and > scholarships: I wanted to beat other boys, and this seemed to be the way in > which I could do so most decisively.

He didn't learn what math was really about till part way through college, when he read Jordan's *Cours d'analyse*.

I shall never forget the astonishment with which I read that remarkable > work, the first inspiration for so many mathematicians of my generation, and > learnt for the first time as I read it what mathematics really meant.

There are two separate kinds of fakeness you need to learn to discount in order to understand what real work is. One is the kind Hardy encountered in school. Subjects get distorted when they're adapted to be taught to kids -- often so distorted that they're nothing like the work done by actual practitioners. 3 The other kind of fakeness is intrinsic to certain types of work. Some types of work are inherently bogus, or at best mere busywork.

There's a kind of solidity to real work. It's not all writing the *Principia* , but it

all feels necessary. That's a vague criterion, but it's deliberately vague, because it has to cover a lot of different types. 4

Once you know the shape of real work, you have to learn how many hours a day to spend on it. You can't solve this problem by simply working every waking hour, because in many kinds of work there's a point beyond which the quality of the result will start to decline.

That limit varies depending on the type of work and the person. I've done several different kinds of work, and the limits were different for each. My limit for the harder types of writing or programming is about five hours a day. Whereas when I was running a startup, I could work all the time. At least for the three years I did it; if I'd kept going much longer, I'd probably have needed to take occasional vacations. 5

The only way to find the limit is by crossing it. Cultivate a sensitivity to the quality of the work you're doing, and then you'll notice if it decreases because you're working too hard. Honesty is critical here, in both directions: you have to notice when you're being lazy, but also when you're working too hard. And if you think there's something admirable about working too hard, get that idea out of your head. You're not merely getting worse results, but getting them because you're showing off -- if not to other people, then to yourself. 6

Finding the limit of working hard is a constant, ongoing process, not something you do just once. Both the difficulty of the work and your ability to do it can vary hour to hour, so you need to be constantly judging both how hard you're trying and how well you're doing.

Trying hard doesn't mean constantly pushing yourself to work, though. There may be some people who do, but I think my experience is fairly typical, and I only have to push myself occasionally when I'm starting a project or when I encounter some sort of check. That's when I'm in danger of procrastinating. But once I get rolling, I tend to keep going.

What keeps me going depends on the type of work. When I was working on Viaweb, I was driven by fear of failure. I barely procrastinated at all then, because there was always something that needed doing, and if I could put more distance between me and the pursuing beast by doing it, why wait? 7 Whereas what drives me now, writing essays, is the flaws in them. Between essays I fuss for a few days, like a dog circling while it decides exactly where to lie down. But once I get started on one, I don't have to push myself to work, because there's always some error or omission already pushing me.

I do make some amount of effort to focus on important topics. Many problems have a hard core at the center, surrounded by easier stuff at the edges. Working hard means aiming toward the center to the extent you can. Some days you may not be able to; some days you'll only be able to work on the easier, peripheral stuff. But you should always be aiming as close to the center as you can without stalling.

The bigger question of what to do with your life is one of these problems with a hard core. There are important problems at the center, which tend to be hard, and less important, easier ones at the edges. So as well as the small, daily adjustments involved in working on a specific problem, you'll occasionally have to make big, lifetime-scale adjustments about which type of work to do. And the rule is the same: working hard means aiming toward the center -- toward the most ambitious problems.

By center, though, I mean the actual center, not merely the current consensus about the center. The consensus about which problems are most important is often mistaken, both in general and within specific fields. If you disagree with it, and you're right, that could represent a valuable opportunity to do something new.

The more ambitious types of work will usually be harder, but although you should not be in denial about this, neither should you treat difficulty as an infallible guide in deciding what to do. If you discover some ambitious type of work that's a bargain in the sense of being easier for you than other people, either because of the abilities you happen to have, or because of some new way you've found to approach it, or simply because you're more excited about it, by all means work on that. Some of the best work is done by people who find an easy way to do something hard.

As well as learning the shape of real work, you need to figure out which kind you're suited for. And that doesn't just mean figuring out which kind your natural abilities match the best; it doesn't mean that if you're 7 feet tall, you have to play basketball. What you're suited for depends not just on your talents but perhaps even more on your interests. A *deep interest* in a topic makes people work harder than any amount of discipline can.

It can be harder to discover your interests than your talents. There are fewer types of talent than interest, and they start to be judged early in childhood, whereas interest in a topic is a subtle thing that may not mature till your twenties, or even later. The topic may not even exist earlier. Plus there are some powerful sources of error you need to learn to discount. Are you really interested in x, or do you want to work on it because you'll make a lot of money, or because other people will be impressed with you, or because your parents want you to? 8

The difficulty of figuring out what to work on varies enormously from one person to another. That's one of the most important things I've learned about work since I was a kid. As a kid, you get the impression that everyone has a calling, and all they have to do is figure out what it is. That's how it works in movies, and in the streamlined biographies fed to kids. Sometimes it works that way in real life. Some people figure out what to do as children and just do it, like Mozart. But others, like Newton, turn restlessly from one kind of work to another. Maybe in retrospect we can identify one as their calling -- we can wish Newton spent more time on math and physics and less on alchemy and theology -- but this is an *illusion* induced by hindsight bias. There was no voice calling to

him that he could have heard.

So while some people's lives converge fast, there will be others whose lives never converge. And for these people, figuring out what to work on is not so much a prelude to working hard as an ongoing part of it, like one of a set of simultaneous equations. For these people, the process I described earlier has a third component: along with measuring both how hard you're working and how well you're doing, you have to think about whether you should keep working in this field or switch to another. If you're working hard but not getting good enough results, you should switch. It sounds simple expressed that way, but in practice it's very difficult. You shouldn't give up on the first day just because you work hard and don't get anywhere. You need to give yourself time to get going. But how much time? And what should you do if work that was going well stops going well? How much time do you give yourself then? ⁹

What even counts as good results? That can be really hard to decide. If you're exploring an area few others have worked in, you may not even know what good results look like. History is full of examples of people who misjudged the importance of what they were working on.

The best test of whether it's worthwhile to work on something is whether you find it interesting. That may sound like a dangerously subjective measure, but it's probably the most accurate one you're going to get. You're the one working on the stuff. Who's in a better position than you to judge whether it's important, and what's a better predictor of its importance than whether it's interesting?

For this test to work, though, you have to be honest with yourself. Indeed, that's the most striking thing about the whole question of working hard: how at each point it depends on being honest with yourself.

Working hard is not just a dial you turn up to 11. It's a complicated, dynamic system that has to be tuned just right at each point. You have to understand the shape of real work, see clearly what kind you're best suited for, aim as close to the true core of it as you can, accurately judge at each moment both what you're capable of and how you're doing, and put in as many hours each day as you can without harming the quality of the result. This network is too complicated to trick. But if you're consistently honest and clear-sighted, it will automatically assume an optimal shape, and you'll be productive in a way few people are.

Notes

[1] In "The Bus Ticket Theory of Genius" I said the three ingredients in great work were natural ability, determination, and interest. That's the formula in the preceding stage; determination and interest yield practice and effort.

[2] I mean this at a resolution of days, not hours. You'll often get somewhere while not working in the sense that the solution to a problem comes to you while taking a *shower*, or even in your sleep, but only because you were working hard on it the day before.

It's good to go on vacation occasionally, but when I go on vacation, I like to learn new things. I wouldn't like just sitting on a beach.

[3] The thing kids do in school that's most like the real version is sports. Admittedly because many sports originated as games played in schools. But in this one area, at least, kids are doing exactly what adults do.

In the average American high school, you have a choice of pretending to do something serious, or seriously doing something pretend. Arguably the latter is no worse.

[4] Knowing what you want to work on doesn't mean you'll be able to. Most people have to spend a lot of their time working on things they don't want to, especially early on. But if you know what you want to do, you at least know what direction to nudge your life in.

[5] The lower time limits for intense work suggest a solution to the problem of having less time to work after you have kids: switch to harder problems. In effect I did that, though not deliberately.

[6] Some cultures have a tradition of performative hard work. I don't love this idea, because (a) it makes a parody of something important and (b) it causes people to wear themselves out doing things that don't matter. I don't know enough to say for sure whether it's net good or bad, but my guess is bad.

[7] One of the reasons people work so hard on startups is that startups can fail, and when they do, that failure tends to be both decisive and conspicuous.

[8] It's ok to work on something to make a lot of money. You need to solve the money problem somehow, and there's nothing wrong with doing that efficiently by trying to make a lot at once. I suppose it would even be ok to be interested in money for its own sake; whatever floats your boat. Just so long as you're conscious of your motivations. The thing to avoid is *unconsciously* letting the need for money warp your ideas about what kind of work you find most interesting.

[9] Many people face this question on a smaller scale with individual projects. But it's easier both to recognize and to accept a dead end in a single project than to abandon some type of work entirely. The more determined you are, the harder it gets. Like a Spanish Flu victim, you're fighting your own immune system: Instead of giving up, you tell yourself, I should just try harder. And who can say you're not right?

Thanks to Trevor Blackwell, John Carmack, John Collison, Patrick Collison, Robert Morris, Geoff Ralston, and Harj Taggar for reading drafts of this.

Arabic Translation

210 Is There Such a Thing as Good Taste?

November 2021

(This essay is derived from a talk at the Cambridge Union.)

When I was a kid, I'd have said there wasn't. My father told me so. Some people like some things, and other people like other things, and who's to say who's right?

It seemed so obvious that there was no such thing as good taste that it was only through indirect evidence that I realized my father was wrong. And that's what I'm going to give you here: a proof by *reductio ad absurdum*. If we start from the premise that there's no such thing as good taste, we end up with conclusions that are obviously false, and therefore the premise must be wrong.

We'd better start by saying what good taste is. There's a narrow sense in which it refers to aesthetic judgements and a broader one in which it refers to preferences of any kind. The strongest proof would be to show that taste exists in the narrowest sense, so I'm going to talk about taste in art. You have better taste than me if the art you like is better than the art I like.

If there's no such thing as good taste, then there's no such thing as *good art*. Because if there is such a thing as good art, it's easy to tell which of two people has better taste. Show them a lot of works by artists they've never seen before and ask them to choose the best, and whoever chooses the better art has better taste.

So if you want to discard the concept of good taste, you also have to discard the concept of good art. And that means you have to discard the possibility of people being good at making it. Which means there's no way for artists to be good at their jobs. And not just visual artists, but anyone who is in any sense an artist. You can't have good actors, or novelists, or composers, or dancers either. You can have popular novelists, but not good ones.

We don't realize how far we'd have to go if we discarded the concept of good taste, because we don't even debate the most obvious cases. But it doesn't just mean we can't say which of two famous painters is better. It means we can't say that any painter is better than a randomly chosen eight year old.

That was how I realized my father was wrong. I started studying painting. And it was just like other kinds of work I'd done: you could do it well, or badly, and if you tried hard, you could get better at it. And it was obvious that Leonardo and Bellini were much better at it than me. That gap between us was not imaginary. They were so good. And if they could be good, then art could be good, and there was such a thing as good taste after all.

Now that I've explained how to show there is such a thing as good taste, I should also explain why people think there isn't. There are two reasons. One is that there's always so much disagreement about taste. Most people's response to

art is a tangle of unexamined impulses. Is the artist famous? Is the subject attractive? Is this the sort of art they're supposed to like? Is it hanging in a famous museum, or reproduced in a big, expensive book? In practice most people's response to art is dominated by such extraneous factors.

And the people who do claim to have good taste are so often mistaken. The paintings admired by the so-called experts in one generation are often so different from those admired a few generations later. It's easy to conclude there's nothing real there at all. It's only when you isolate this force, for example by trying to paint and comparing your work to Bellini's, that you can see that it does in fact exist.

The other reason people doubt that art can be good is that there doesn't seem to be any room in the art for this goodness. The argument goes like this. Imagine several people looking at a work of art and judging how good it is. If being good art really is a property of objects, it should be in the object somehow. But it doesn't seem to be; it seems to be something happening in the heads of each of the observers. And if they disagree, how do you choose between them?

The solution to this puzzle is to realize that the purpose of art is to work on its human audience, and humans have a lot in common. And to the extent the things an object acts upon respond in the same way, that's arguably what it means for the object to have the corresponding property. If everything a particle interacts with behaves as if the particle had a mass of m , then it has a mass of m . So the distinction between "objective" and "subjective" is not binary, but a matter of degree, depending on how much the subjects have in common. Particles interacting with one another are at one pole, but people interacting with art are not all the way at the other; their reactions aren't *random*.

Because people's responses to art aren't random, art can be designed to operate on people, and be good or bad depending on how effectively it does so. Much as a vaccine can be. If someone were talking about the ability of a vaccine to confer immunity, it would seem very frivolous to object that conferring immunity wasn't really a property of vaccines, because acquiring immunity is something that happens in the immune system of each individual person. Sure, people's immune systems vary, and a vaccine that worked on one might not work on another, but that doesn't make it meaningless to talk about the effectiveness of a vaccine.

The situation with art is messier, of course. You can't measure effectiveness by simply taking a vote, as you do with vaccines. You have to imagine the responses of subjects with a deep knowledge of art, and enough clarity of mind to be able to ignore extraneous influences like the fame of the artist. And even then you'd still see some disagreement. People do vary, and judging art is hard, especially recent art. There is definitely not a total order either of works or of people's ability to judge them. But there is equally definitely a partial order of both. So while it's not possible to have perfect taste, it is possible to have good taste.

Thanks to the Cambridge Union for inviting me, and to Trevor Blackwell,

Jessica Livingston, and Robert Morris for reading drafts of this.

216 The Need to Read

November 2022

In the science fiction books I read as a kid, reading had often been replaced by some more efficient way of acquiring knowledge. Mysterious "tapes" would load it into one's brain like a program being loaded into a computer.

That sort of thing is unlikely to happen anytime soon. Not just because it would be hard to build a replacement for reading, but because even if one existed, it would be insufficient. Reading about x doesn't just teach you about x ; it also teaches you how to write. ¹

Would that matter? If we replaced reading, would anyone need to be good at writing?

The reason it would matter is that writing is not just a way to convey ideas, but also a way to have them.

A good writer doesn't just think, and then write down what he thought, as a sort of transcript. A good writer will almost always discover new things in the process of writing. And there is, as far as I know, no substitute for this kind of discovery. Talking about your ideas with other people is a good way to develop them. But even after doing this, you'll find you still discover new things when you sit down to write. There is a kind of thinking that can only be done by *writing*.

There are of course kinds of thinking that can be done without writing. If you don't need to go too deeply into a problem, you can solve it without writing. If you're thinking about how two pieces of machinery should fit together, writing about it probably won't help much. And when a problem can be described formally, you can sometimes solve it in your head. But if you need to solve a complicated, ill-defined problem, it will almost always help to write about it. Which in turn means that someone who's not good at writing will almost always be at a disadvantage in solving such problems.

You can't think well without writing well, and you can't write well without reading well. And I mean that last "well" in both senses. You have to be good at reading, and read good things. ²

People who just want information may find other ways to get it. But people who want to have ideas can't afford to.

Notes

[1] Audiobooks can give you examples of good writing, but having them read to you doesn't teach you as much about writing as reading them yourself.

[2] By "good at reading" I don't mean good at the mechanics of reading. You don't have to be good at extracting words from the page so much as extracting meaning from the words.

Japanese Translation Chinese Translation

Italian Translation French Translation

217 How to Get New Ideas

January 2023

(Someone fed my essays into GPT to make something that could answer questions based on them, then asked it where good ideas come from. The answer was ok, but not what I would have said. This is what I would have said.)

The way to get new ideas is to notice anomalies: what seems strange, or missing, or broken? You can see anomalies in everyday life (much of standup comedy is based on this), but the best place to look for them is at the frontiers of knowledge.

Knowledge grows fractally. From a distance its edges look smooth, but when you learn enough to get close to one, you'll notice it's full of gaps. These gaps will seem obvious; it will seem inexplicable that no one has tried x or wondered about y. In the best case, exploring such gaps yields whole new fractal buds.

218 How to Do Great Work

July 2023

If you collected lists of techniques for doing great work in a lot of different fields, what would the intersection look like? I decided to find out by making it.

Partly my goal was to create a guide that could be used by someone working in any field. But I was also curious about the shape of the intersection. And one thing this exercise shows is that it does have a definite shape; it's not just a point labelled "work hard."

The following recipe assumes you're very ambitious.

The first step is to decide what to work on. The work you choose needs to have three qualities: it has to be something you have a natural aptitude for, that you have a deep interest in, and that offers scope to do great work.

In practice you don't have to worry much about the third criterion. Ambitious people are if anything already too conservative about it. So all you need to do is find something you have an aptitude for and great interest in. 1

That sounds straightforward, but it's often quite difficult. When you're young you don't know what you're good at or what different kinds of work are like. Some kinds of work you end up doing may not even exist yet. So while some people know what they want to do at 14, most have to figure it out.

The way to figure out what to work on is by working. If you're not sure what to work on, guess. But pick something and get going. You'll probably guess wrong some of the time, but that's fine. It's good to know about multiple things; some of the biggest discoveries come from noticing connections between different fields.

Develop a habit of working on your own projects. Don't let "work" mean something other people tell you to do. If you do manage to do great work one day, it will probably be on a project of your own. It may be within some bigger project, but you'll be driving your part of it.

What should your projects be? Whatever seems to you excitingly ambitious. As you grow older and your taste in projects evolves, exciting and important will converge. At 7 it may seem excitingly ambitious to build huge things out of Lego, then at 14 to teach yourself calculus, till at 21 you're starting to explore unanswered questions in physics. But always preserve excitingness.

There's a kind of excited curiosity that's both the engine and the rudder of great work. It will not only drive you, but if you let it have its way, will also show you what to work on.

What are you excessively curious about -- curious to a degree that would bore most other people? That's what you're looking for.

Once you've found something you're excessively interested in, the next step is to learn enough about it to get you to one of the frontiers of knowledge. Knowledge expands fractally, and from a distance its edges look smooth, but once you learn enough to get close to one, they turn out to be full of gaps.

The next step is to notice them. This takes some skill, because your brain wants to ignore such gaps in order to make a simpler model of the world. Many discoveries have come from asking questions about things that everyone else took for granted. 2

If the answers seem strange, so much the better. Great work often has a tincture of strangeness. You see this from painting to math. It would be affected to try to manufacture it, but if it appears, embrace it.

Boldly chase outlier ideas, even if other people aren't interested in them -- in fact, especially if they aren't. If you're excited about some possibility that everyone else ignores, and you have enough expertise to say precisely what they're all overlooking, that's as good a bet as you'll find. 3

Four steps: choose a field, learn enough to get to the frontier, notice gaps, explore promising ones. This is how practically everyone who's done great work has done it, from painters to physicists.

Steps two and four will require hard work. It may not be possible to prove that you have to work hard to do great things, but the empirical evidence is on the scale of the evidence for mortality. That's why it's essential to work on something you're deeply interested in. Interest will drive you to work harder than mere diligence ever could.

The three most powerful motives are curiosity, delight, and the desire to do something impressive. Sometimes they converge, and that combination is the most powerful of all.

The big prize is to discover a new fractal bud. You notice a crack in the surface of knowledge, pry it open, and there's a whole world inside.

Let's talk a little more about the complicated business of figuring out what to work on. The main reason it's hard is that you can't tell what most kinds of work are like except by doing them. Which means the four steps overlap: you may have to work at something for years before you know how much you like it or how good you are at it. And in the meantime you're not doing, and thus not learning about, most other kinds of work. So in the worst case you choose late based on very incomplete information. 4

The nature of ambition exacerbates this problem. Ambition comes in two forms, one that precedes interest in the subject and one that grows out of it. Most people who do great work have a mix, and the more you have of the former, the harder it will be to decide what to do.

The educational systems in most countries pretend it's easy. They expect you to commit to a field long before you could know what it's really like. And as a result an ambitious person on an optimal trajectory will often read to the system as an instance of breakage.

It would be better if they at least admitted it -- if they admitted that the system not only can't do much to help you figure out what to work on, but is designed on the assumption that you'll somehow magically guess as a teenager. They don't tell you, but I will: when it comes to figuring out what to work on, you're on your own. Some people get lucky and do guess correctly, but the rest will find themselves scrambling diagonally across tracks laid down on the assumption that everyone does.

What should you do if you're young and ambitious but don't know what to work on? What you should not do is drift along passively, assuming the problem will solve itself. You need to take action. But there is no systematic procedure you can follow. When you read biographies of people who've done great work, it's remarkable how much luck is involved. They discover what to work on as a result of a chance meeting, or by reading a book they happen to pick up. So you need to make yourself a big target for luck, and the way to do that is to be

curious. Try lots of things, meet lots of people, read lots of books, ask lots of questions. 5

When in doubt, optimize for interestingness. Fields change as you learn more about them. What mathematicians do, for example, is very different from what you do in high school math classes. So you need to give different types of work a chance to show you what they're like. But a field should become *increasingly* interesting as you learn more about it. If it doesn't, it's probably not for you.

Don't worry if you find you're interested in different things than other people. The stranger your tastes in interestingness, the better. Strange tastes are often strong ones, and a strong taste for work means you'll be productive. And you're more likely to find new things if you're looking where few have looked before.

One sign that you're suited for some kind of work is when you like even the parts that other people find tedious or frightening.

But fields aren't people; you don't owe them any loyalty. If in the course of working on one thing you discover another that's more exciting, don't be afraid to switch.

If you're making something for people, make sure it's something they actually want. The best way to do this is to make something you yourself want. Write the story you want to read; build the tool you want to use. Since your friends probably have similar interests, this will also get you your initial audience.

This *should* follow from the excitingness rule. Obviously the most exciting story to write will be the one you want to read. The reason I mention this case explicitly is that so many people get it wrong. Instead of making what they want, they try to make what some imaginary, more sophisticated audience wants. And once you go down that route, you're lost. 6

There are a lot of forces that will lead you astray when you're trying to figure out what to work on. Pretentiousness, fashion, fear, money, politics, other people's wishes, eminent frauds. But if you stick to what you find genuinely interesting, you'll be proof against all of them. If you're interested, you're not astray.

Following your interests may sound like a rather passive strategy, but in practice it usually means following them past all sorts of obstacles. You usually have to risk rejection and failure. So it does take a good deal of boldness.

But while you need boldness, you don't usually need much planning. In most cases the recipe for doing great work is simply: work hard on excitingly ambitious projects, and something good will come of it. Instead of making a plan and then executing it, you just try to preserve certain invariants.

The trouble with planning is that it only works for achievements you can describe in advance. You can win a gold medal or get rich by deciding to as a child and then tenaciously pursuing that goal, but you can't discover natural selection that way.

I think for most people who want to do great work, the right strategy is not to plan too much. At each stage do whatever seems most interesting and gives you the best options for the future. I call this approach "staying upwind." This is how most people who've done great work seem to have done it.

Even when you've found something exciting to work on, working on it is not always straightforward. There will be times when some new idea makes you leap out of bed in the morning and get straight to work. But there will also be plenty of times when things aren't like that.

You don't just put out your sail and get blown forward by inspiration. There are headwinds and currents and hidden shoals. So there's a technique to working, just as there is to sailing.

For example, while you must work hard, it's possible to work too hard, and if you do that you'll find you get diminishing returns: fatigue will make you stupid, and eventually even damage your health. The point at which work yields diminishing returns depends on the type. Some of the hardest types you might only be able to do for four or five hours a day.

Ideally those hours will be contiguous. To the extent you can, try to arrange your life so you have big blocks of time to work in. You'll shy away from hard tasks if you know you might be interrupted.

It will probably be harder to start working than to keep working. You'll often have to trick yourself to get over that initial threshold. Don't worry about this; it's the nature of work, not a flaw in your character. Work has a sort of activation energy, both per day and per project. And since this threshold is fake in the sense that it's higher than the energy required to keep going, it's ok to tell yourself a lie of corresponding magnitude to get over it.

It's usually a mistake to lie to yourself if you want to do great work, but this is one of the rare cases where it isn't. When I'm reluctant to start work in the morning, I often trick myself by saying "I'll just read over what I've got so far." Five minutes later I've found something that seems mistaken or incomplete, and I'm off.

Similar techniques work for starting new projects. It's ok to lie to yourself about how much work a project will entail, for example. Lots of great things began with someone saying "How hard could it be?"

This is one case where the young have an advantage. They're more optimistic, and even though one of the sources of their optimism is ignorance, in this case ignorance can sometimes beat knowledge.

Try to finish what you start, though, even if it turns out to be more work than you expected. Finishing things is not just an exercise in tidiness or self-discipline. In many projects a lot of the best work happens in what was meant to be the final stage.

Another permissible lie is to exaggerate the importance of what you're working on, at least in your own mind. If that helps you discover something new, it may turn out not to have been a lie after all. 7

Since there are two senses of starting work -- per day and per project -- there are also two forms of procrastination. Per-project procrastination is far the more dangerous. You put off starting that ambitious project from year to year because the time isn't quite right. When you're procrastinating in units of years, you can get a lot not done. 8

One reason per-project procrastination is so dangerous is that it usually camouflages itself as work. You're not just sitting around doing nothing; you're working industriously on something else. So per-project procrastination doesn't set off the alarms that per-day procrastination does. You're too busy to notice it.

The way to beat it is to stop occasionally and ask yourself: Am I working on what I most want to work on? When you're young it's ok if the answer is sometimes no, but this gets increasingly dangerous as you get older. 9

Great work usually entails spending what would seem to most people an unreasonable amount of time on a problem. You can't think of this time as a cost, or it will seem too high. You have to find the work sufficiently engaging as it's happening.

There may be some jobs where you have to work diligently for years at things you hate before you get to the good part, but this is not how great work happens. Great work happens by focusing consistently on something you're genuinely interested in. When you pause to take stock, you're surprised how far you've come.

The reason we're surprised is that we underestimate the cumulative effect of work. Writing a page a day doesn't sound like much, but if you do it every day you'll write a book a year. That's the key: consistency. People who do great things don't get a lot done every day. They get something done, rather than nothing.

If you do work that compounds, you'll get exponential growth. Most people who do this do it unconsciously, but it's worth stopping to think about. Learning, for example, is an instance of this phenomenon: the more you learn about something, the easier it is to learn more. Growing an audience is another: the more fans you have, the more new fans they'll bring you.

The trouble with exponential growth is that the curve feels flat in the beginning. It isn't; it's still a wonderful exponential curve. But we can't grasp that intuitively, so we underrate exponential growth in its early stages.

Something that grows exponentially can become so valuable that it's worth making an extraordinary effort to get it started. But since we underrate exponential growth early on, this too is mostly done unconsciously: people push through

the initial, unrewarding phase of learning something new because they know from experience that learning new things always takes an initial push, or they grow their audience one fan at a time because they have nothing better to do. If people consciously realized they could invest in exponential growth, many more would do it.

Work doesn't just happen when you're trying to. There's a kind of undirected thinking you do when walking or taking a shower or lying in bed that can be very powerful. By letting your mind wander a little, you'll often solve problems you were unable to solve by frontal attack.

You have to be working hard in the normal way to benefit from this phenomenon, though. You can't just walk around daydreaming. The daydreaming has to be interleaved with deliberate work that feeds it questions. 10

Everyone knows to avoid distractions at work, but it's also important to avoid them in the other half of the cycle. When you let your mind wander, it wanders to whatever you care about most at that moment. So avoid the kind of distraction that pushes your work out of the top spot, or you'll waste this valuable type of thinking on the distraction instead. (Exception: Don't avoid love.)

Consciously cultivate your taste in the work done in your field. Until you know which is the best and what makes it so, you don't know what you're aiming for.

And that *is* what you're aiming for, because if you don't try to be the best, you won't even be good. This observation has been made by so many people in so many different fields that it might be worth thinking about why it's true. It could be because ambition is a phenomenon where almost all the error is in one direction -- where almost all the shells that miss the target miss by falling short. Or it could be because ambition to be the best is a qualitatively different thing from ambition to be good. Or maybe being good is simply too vague a standard. Probably all three are true. 11

Fortunately there's a kind of economy of scale here. Though it might seem like you'd be taking on a heavy burden by trying to be the best, in practice you often end up net ahead. It's exciting, and also strangely liberating. It simplifies things. In some ways it's easier to try to be the best than to try merely to be good.

One way to aim high is to try to make something that people will care about in a hundred years. Not because their opinions matter more than your contemporaries', but because something that still seems good in a hundred years is more likely to be genuinely good.

Don't try to work in a distinctive style. Just try to do the best job you can; you won't be able to help doing it in a distinctive way.

Style is doing things in a distinctive way without trying to. Trying to is affectation.

Affectation is in effect to pretend that someone other than you is doing the work. You adopt an impressive but fake persona, and while you're pleased with the impressiveness, the fakeness is what shows in the work. 12

The temptation to be someone else is greatest for the young. They often feel like nobodies. But you never need to worry about that problem, because it's self-solving if you work on sufficiently ambitious projects. If you succeed at an ambitious project, you're not a nobody; you're the person who did it. So just do the work and your identity will take care of itself.

"Avoid affectation" is a useful rule so far as it goes, but how would you express this idea positively? How would you say what to be, instead of what not to be? The best answer is earnest. If you're earnest you avoid not just affectation but a whole set of similar vices.

The core of being earnest is being intellectually honest. We're taught as children to be honest as an unselfish virtue -- as a kind of sacrifice. But in fact it's a source of power too. To see new ideas, you need an exceptionally sharp eye for the truth. You're trying to see more truth than others have seen so far. And how can you have a sharp eye for the truth if you're intellectually dishonest?

One way to avoid intellectual dishonesty is to maintain a slight positive pressure in the opposite direction. Be aggressively willing to admit that you're mistaken. Once you've admitted you were mistaken about something, you're free. Till then you have to carry it. 13

Another more subtle component of earnestness is informality. Informality is much more important than its grammatically negative name implies. It's not merely the absence of something. It means focusing on what matters instead of what doesn't.

What formality and affectation have in common is that as well as doing the work, you're trying to seem a certain way as you're doing it. But any energy that goes into how you seem comes out of being good. That's one reason nerds have an advantage in doing great work: they expend little effort on seeming anything. In fact that's basically the definition of a nerd.

Nerds have a kind of innocent boldness that's exactly what you need in doing great work. It's not learned; it's preserved from childhood. So hold onto it. Be the one who puts things out there rather than the one who sits back and offers sophisticated-sounding criticisms of them. "It's easy to criticize" is true in the most literal sense, and the route to great work is never easy.

There may be some jobs where it's an advantage to be cynical and pessimistic, but if you want to do great work it's an advantage to be optimistic, even though that means you'll risk looking like a fool sometimes. There's an old tradition of doing the opposite. The Old Testament says it's better to keep quiet lest you look like a fool. But that's advice for *seeming* smart. If you actually want to discover new things, it's better to take the risk of telling people your ideas.

Some people are naturally earnest, and with others it takes a conscious effort. Either kind of earnestness will suffice. But I doubt it would be possible to do great work without being earnest. It's so hard to do even if you are. You don't have enough margin for error to accommodate the distortions introduced by being affected, intellectually dishonest, orthodox, fashionable, or cool. 14

Great work is consistent not only with who did it, but with itself. It's usually all of a piece. So if you face a decision in the middle of working on something, ask which choice is more consistent.

You may have to throw things away and redo them. You won't necessarily have to, but you have to be willing to. And that can take some effort; when there's something you need to redo, status quo bias and laziness will combine to keep you in denial about it. To beat this ask: If I'd already made the change, would I want to revert to what I have now?

Have the confidence to cut. Don't keep something that doesn't fit just because you're proud of it, or because it cost you a lot of effort.

Indeed, in some kinds of work it's good to strip whatever you're doing to its essence. The result will be more concentrated; you'll understand it better; and you won't be able to lie to yourself about whether there's anything real there.

Mathematical elegance may sound like a mere metaphor, drawn from the arts. That's what I thought when I first heard the term "elegant" applied to a proof. But now I suspect it's conceptually prior -- that the main ingredient in artistic elegance is mathematical elegance. At any rate it's a useful standard well beyond math.

Elegance can be a long-term bet, though. Laborious solutions will often have more prestige in the short term. They cost a lot of effort and they're hard to understand, both of which impress people, at least temporarily.

Whereas some of the very best work will seem like it took comparatively little effort, because it was in a sense already there. It didn't have to be built, just seen. It's a very good sign when it's hard to say whether you're creating something or discovering it.

When you're doing work that could be seen as either creation or discovery, err on the side of discovery. Try thinking of yourself as a mere conduit through which the ideas take their natural shape.

(Strangely enough, one exception is the problem of choosing a problem to work on. This is usually seen as search, but in the best case it's more like creating something. In the best case you create the field in the process of exploring it.)

Similarly, if you're trying to build a powerful tool, make it gratuitously unrestrictive. A powerful tool almost by definition will be used in ways you didn't expect, so err on the side of eliminating restrictions, even if you don't know what the benefit will be.

Great work will often be tool-like in the sense of being something others build on. So it's a good sign if you're creating ideas that others could use, or exposing questions that others could answer. The best ideas have implications in many different areas.

If you express your ideas in the most general form, they'll be truer than you intended.

True by itself is not enough, of course. Great ideas have to be true and new. And it takes a certain amount of ability to see new ideas even once you've learned enough to get to one of the frontiers of knowledge.

In English we give this ability names like originality, creativity, and imagination. And it seems reasonable to give it a separate name, because it does seem to some extent a separate skill. It's possible to have a great deal of ability in other respects -- to have a great deal of what's often called "technical ability" -- and yet not have much of this.

I've never liked the term "creative process." It seems misleading. Originality isn't a process, but a habit of mind. Original thinkers throw off new ideas about whatever they focus on, like an angle grinder throwing off sparks. They can't help it.

If the thing they're focused on is something they don't understand very well, these new ideas might not be good. One of the most original thinkers I know decided to focus on dating after he got divorced. He knew roughly as much about dating as the average 15 year old, and the results were spectacularly colorful. But to see originality separated from expertise like that made its nature all the more clear.

I don't know if it's possible to cultivate originality, but there are definitely ways to make the most of however much you have. For example, you're much more likely to have original ideas when you're working on something. Original ideas don't come from trying to have original ideas. They come from trying to build or understand something slightly too difficult. 15

Talking or writing about the things you're interested in is a good way to generate new ideas. When you try to put ideas into words, a missing idea creates a sort of vacuum that draws it out of you. Indeed, there's a kind of thinking that can only be done by writing.

Changing your context can help. If you visit a new place, you'll often find you have new ideas there. The journey itself often dislodges them. But you may not have to go far to get this benefit. Sometimes it's enough just to go for a walk. 16

It also helps to travel in topic space. You'll have more new ideas if you explore lots of different topics, partly because it gives the angle grinder more surface area to work on, and partly because analogies are an especially fruitful source of new ideas.

Don't divide your attention *evenly* between many topics though, or you'll spread yourself too thin. You want to distribute it according to something more like a power law. 17 Be professionally curious about a few topics and idly curious about many more.

Curiosity and originality are closely related. Curiosity feeds originality by giving it new things to work on. But the relationship is closer than that. Curiosity is itself a kind of originality; it's roughly to questions what originality is to answers. And since questions at their best are a big component of answers, curiosity at its best is a creative force.

Having new ideas is a strange game, because it usually consists of seeing things that were right under your nose. Once you've seen a new idea, it tends to seem obvious. Why did no one think of this before?

When an idea seems simultaneously novel and obvious, it's probably a good one.

Seeing something obvious sounds easy. And yet empirically having new ideas is hard. What's the source of this apparent contradiction? It's that seeing the new idea usually requires you to change the way you look at the world. We see the world through models that both help and constrain us. When you fix a broken model, new ideas become obvious. But noticing and fixing a broken model is hard. That's how new ideas can be both obvious and yet hard to discover: they're easy to see after you do something hard.

One way to discover broken models is to be stricter than other people. Broken models of the world leave a trail of clues where they bash against reality. Most people don't want to see these clues. It would be an understatement to say that they're attached to their current model; it's what they think in; so they'll tend to ignore the trail of clues left by its breakage, however conspicuous it may seem in retrospect.

To find new ideas you have to seize on signs of breakage instead of looking away. That's what Einstein did. He was able to see the wild implications of Maxwell's equations not so much because he was looking for new ideas as because he was stricter.

The other thing you need is a willingness to break rules. Paradoxical as it sounds, if you want to fix your model of the world, it helps to be the sort of person who's comfortable breaking rules. From the point of view of the old model, which everyone including you initially shares, the new model usually breaks at least implicit rules.

Few understand the degree of rule-breaking required, because new ideas seem much more conservative once they succeed. They seem perfectly reasonable once you're using the new model of the world they brought with them. But they didn't at the time; it took the greater part of a century for the heliocentric model to be generally accepted, even among astronomers, because it felt so wrong.

Indeed, if you think about it, a good new idea has to seem bad to most people,

or someone would have already explored it. So what you're looking for is ideas that seem crazy, but the right kind of crazy. How do you recognize these? You can't with certainty. Often ideas that seem bad are bad. But ideas that are the right kind of crazy tend to be exciting; they're rich in implications; whereas ideas that are merely bad tend to be depressing.

There are two ways to be comfortable breaking rules: to enjoy breaking them, and to be indifferent to them. I call these two cases being aggressively and passively independent-minded.

The aggressively independent-minded are the naughty ones. Rules don't merely fail to stop them; breaking rules gives them additional energy. For this sort of person, delight at the sheer audacity of a project sometimes supplies enough activation energy to get it started.

The other way to break rules is not to care about them, or perhaps even to know they exist. This is why novices and outsiders often make new discoveries; their ignorance of a field's assumptions acts as a source of temporary passive independent-mindedness. Aspies also seem to have a kind of immunity to conventional beliefs. Several I know say that this helps them to have new ideas.

Strictness plus rule-breaking sounds like a strange combination. In popular culture they're opposed. But popular culture has a broken model in this respect. It implicitly assumes that issues are trivial ones, and in trivial matters strictness and rule-breaking *are* opposed. But in questions that really matter, only rule-breakers can be truly strict.

An overlooked idea often doesn't lose till the semifinals. You do see it, subconsciously, but then another part of your subconscious shoots it down because it would be too weird, too risky, too much work, too controversial. This suggests an exciting possibility: if you could turn off such filters, you could see more new ideas.

One way to do that is to ask what would be good ideas for *someone else* to explore. Then your subconscious won't shoot them down to protect you.

You could also discover overlooked ideas by working in the other direction: by starting from what's obscuring them. Every cherished but mistaken principle is surrounded by a dead zone of valuable ideas that are unexplored because they contradict it.

Religions are collections of cherished but mistaken principles. So anything that can be described either literally or metaphorically as a religion will have valuable unexplored ideas in its shadow. Copernicus and Darwin both made discoveries of this type. 18

What are people in your field religious about, in the sense of being too attached to some principle that might not be as self-evident as they think? What becomes possible if you discard it?

People show much more originality in solving problems than in deciding which problems to solve. Even the smartest can be surprisingly conservative when deciding what to work on. People who'd never dream of being fashionable in any other way get sucked into working on fashionable problems.

One reason people are more conservative when choosing problems than solutions is that problems are bigger bets. A problem could occupy you for years, while exploring a solution might only take days. But even so I think most people are too conservative. They're not merely responding to risk, but to fashion as well. Unfashionable problems are undervalued.

One of the most interesting kinds of unfashionable problem is the problem that people think has been fully explored, but hasn't. Great work often takes something that already exists and shows its latent potential. Durer and Watt both did this. So if you're interested in a field that others think is tapped out, don't let their skepticism deter you. People are often wrong about this.

Working on an unfashionable problem can be very pleasing. There's no hype or hurry. Opportunists and critics are both occupied elsewhere. The existing work often has an old-school solidity. And there's a satisfying sense of economy in cultivating ideas that would otherwise be wasted.

But the most common type of overlooked problem is not explicitly unfashionable in the sense of being out of fashion. It just doesn't seem to matter as much as it actually does. How do you find these? By being self-indulgent -- by letting your curiosity have its way, and tuning out, at least temporarily, the little voice in your head that says you should only be working on "important" problems.

You do need to work on important problems, but almost everyone is too conservative about what counts as one. And if there's an important but overlooked problem in your neighborhood, it's probably already on your subconscious radar screen. So try asking yourself: if you were going to take a break from "serious" work to work on something just because it would be really interesting, what would you do? The answer is probably more important than it seems.

Originality in choosing problems seems to matter even more than originality in solving them. That's what distinguishes the people who discover whole new fields. So what might seem to be merely the initial step -- deciding what to work on -- is in a sense the key to the whole game.

Few grasp this. One of the biggest misconceptions about new ideas is about the ratio of question to answer in their composition. People think big ideas are answers, but often the real insight was in the question.

Part of the reason we underrate questions is the way they're used in schools. In schools they tend to exist only briefly before being answered, like unstable particles. But a really good question can be much more than that. A really good question is a partial discovery. How do new species arise? Is the force that makes objects fall to earth the same as the one that keeps planets in their orbits? By even asking such questions you were already in excitingly novel territory.

Unanswered questions can be uncomfortable things to carry around with you. But the more you're carrying, the greater the chance of noticing a solution -- or perhaps even more excitingly, noticing that two unanswered questions are the same.

Sometimes you carry a question for a long time. Great work often comes from returning to a question you first noticed years before -- in your childhood, even -- and couldn't stop thinking about. People talk a lot about the importance of keeping your youthful dreams alive, but it's just as important to keep your youthful questions alive. 19

This is one of the places where actual expertise differs most from the popular picture of it. In the popular picture, experts are certain. But actually the more puzzled you are, the better, so long as (a) the things you're puzzled about matter, and (b) no one else understands them either.

Think about what's happening at the moment just before a new idea is discovered. Often someone with sufficient expertise is puzzled about something. Which means that originality consists partly of puzzlement -- of confusion! You have to be comfortable enough with the world being full of puzzles that you're willing to see them, but not so comfortable that you don't want to solve them. 20

It's a great thing to be rich in unanswered questions. And this is one of those situations where the rich get richer, because the best way to acquire new questions is to try answering existing ones. Questions don't just lead to answers, but also to more questions.

The best questions grow in the answering. You notice a thread protruding from the current paradigm and try pulling on it, and it just gets longer and longer. So don't require a question to be obviously big before you try answering it. You can rarely predict that. It's hard enough even to notice the thread, let alone to predict how much will unravel if you pull on it.

It's better to be promiscuously curious -- to pull a little bit on a lot of threads, and see what happens. Big things start small. The initial versions of big things were often just experiments, or side projects, or talks, which then grew into something bigger. So start lots of small things.

Being prolific is underrated. The more different things you try, the greater the chance of discovering something new. Understand, though, that trying lots of things will mean trying lots of things that don't work. You can't have a lot of good ideas without also having a lot of bad ones. 21

Though it sounds more responsible to begin by studying everything that's been done before, you'll learn faster and have more fun by trying stuff. And you'll understand previous work better when you do look at it. So err on the side of starting. Which is easier when starting means starting small; those two ideas fit together like two puzzle pieces.

How do you get from starting small to doing something great? By making

successive versions. Great things are almost always made in successive versions. You start with something small and evolve it, and the final version is both cleverer and more ambitious than anything you could have planned.

It's particularly useful to make successive versions when you're making something for people -- to get an initial version in front of them quickly, and then evolve it based on their response.

Begin by trying the simplest thing that could possibly work. Surprisingly often, it does. If it doesn't, this will at least get you started.

Don't try to cram too much new stuff into any one version. There are names for doing this with the first version (taking too long to ship) and the second (the second system effect), but these are both merely instances of a more general principle.

An early version of a new project will sometimes be dismissed as a toy. It's a good sign when people do this. That means it has everything a new idea needs except scale, and that tends to follow. 22

The alternative to starting with something small and evolving it is to plan in advance what you're going to do. And planning does usually seem the more responsible choice. It sounds more organized to say "we're going to do x and then y and then z" than "we're going to try x and see what happens." And it is more *organized* ; it just doesn't work as well.

Planning per se isn't good. It's sometimes necessary, but it's a necessary evil -- a response to unforgiving conditions. It's something you have to do because you're working with inflexible media, or because you need to coordinate the efforts of a lot of people. If you keep projects small and use flexible media, you don't have to plan as much, and your designs can evolve instead.

Take as much risk as you can afford. In an efficient market, risk is proportionate to reward, so don't look for certainty, but for a bet with high expected value. If you're not failing occasionally, you're probably being too conservative.

Though conservatism is usually associated with the old, it's the young who tend to make this mistake. Inexperience makes them fear risk, but it's when you're young that you can afford the most.

Even a project that fails can be valuable. In the process of working on it, you'll have crossed territory few others have seen, and encountered questions few others have asked. And there's probably no better source of questions than the ones you encounter in trying to do something slightly too hard.

Use the advantages of youth when you have them, and the advantages of age once you have those. The advantages of youth are energy, time, optimism, and freedom. The advantages of age are knowledge, efficiency, money, and power. With effort you can acquire some of the latter when young and keep some of the former when old.

The old also have the advantage of knowing which advantages they have. The young often have them without realizing it. The biggest is probably time. The young have no idea how rich they are in time. The best way to turn this time to advantage is to use it in slightly frivolous ways: to learn about something you don't need to know about, just out of curiosity, or to try building something just because it would be cool, or to become freakishly good at something.

That "slightly" is an important qualification. Spend time lavishly when you're young, but don't simply waste it. There's a big difference between doing something you worry might be a waste of time and doing something you know for sure will be. The former is at least a bet, and possibly a better one than you think. 23

The most subtle advantage of youth, or more precisely of inexperience, is that you're seeing everything with fresh eyes. When your brain embraces an idea for the first time, sometimes the two don't fit together perfectly. Usually the problem is with your brain, but occasionally it's with the idea. A piece of it sticks out awkwardly and jabs you when you think about it. People who are used to the idea have learned to ignore it, but you have the opportunity not to. 24

So when you're learning about something for the first time, pay attention to things that seem wrong or missing. You'll be tempted to ignore them, since there's a 99% chance the problem is with you. And you may have to set aside your misgivings temporarily to keep progressing. But don't forget about them. When you've gotten further into the subject, come back and check if they're still there. If they're still viable in the light of your present knowledge, they probably represent an undiscovered idea.

One of the most valuable kinds of knowledge you get from experience is to know what you *don't* have to worry about. The young know all the things that could matter, but not their relative importance. So they worry equally about everything, when they should worry much more about a few things and hardly at all about the rest.

But what you don't know is only half the problem with inexperience. The other half is what you do know that ain't so. You arrive at adulthood with your head full of nonsense -- bad habits you've acquired and false things you've been taught -- and you won't be able to do great work till you clear away at least the nonsense in the way of whatever type of work you want to do.

Much of the nonsense left in your head is left there by schools. We're so used to schools that we unconsciously treat going to school as identical with learning, but in fact schools have all sorts of strange qualities that warp our ideas about learning and thinking.

For example, schools induce passivity. Since you were a small child, there was an authority at the front of the class telling all of you what you had to learn and then measuring whether you did. But neither classes nor tests are intrinsic

to learning; they're just artifacts of the way schools are usually designed.

The sooner you overcome this passivity, the better. If you're still in school, try thinking of your education as your project, and your teachers as working for you rather than vice versa. That may seem a stretch, but it's not merely some weird thought experiment. It's the truth, economically, and in the best case it's the truth intellectually as well. The best teachers don't want to be your bosses. They'd prefer it if you pushed ahead, using them as a source of advice, rather than being pulled by them through the material.

Schools also give you a misleading impression of what work is like. In school they tell you what the problems are, and they're almost always soluble using no more than you've been taught so far. In real life you have to figure out what the problems are, and you often don't know if they're soluble at all.

But perhaps the worst thing schools do to you is train you to win by hacking the test. You can't do great work by doing that. You can't trick God. So stop looking for that kind of shortcut. The way to beat the system is to focus on problems and solutions that others have overlooked, not to skimp on the work itself.

Don't think of yourself as dependent on some gatekeeper giving you a "big break." Even if this were true, the best way to get it would be to focus on doing good work rather than chasing influential people.

And don't take rejection by committees to heart. The qualities that impress admissions officers and prize committees are quite different from those required to do great work. The decisions of selection committees are only meaningful to the extent that they're part of a feedback loop, and very few are.

People new to a field will often copy existing work. There's nothing inherently bad about that. There's no better way to learn how something works than by trying to reproduce it. Nor does copying necessarily make your work unoriginal. Originality is the presence of new ideas, not the absence of old ones.

There's a good way to copy and a bad way. If you're going to copy something, do it openly instead of furtively, or worse still, unconsciously. This is what's meant by the famously misattributed phrase "Great artists steal." The really dangerous kind of copying, the kind that gives copying a bad name, is the kind that's done without realizing it, because you're nothing more than a train running on tracks laid down by someone else. But at the other extreme, copying can be a sign of superiority rather than subordination. 25

In many fields it's almost inevitable that your early work will be in some sense based on other people's. Projects rarely arise in a vacuum. They're usually a reaction to previous work. When you're first starting out, you don't have any previous work; if you're going to react to something, it has to be someone else's. Once you're established, you can react to your own. But while the former gets called derivative and the latter doesn't, structurally the two cases are more similar than they seem.

Oddly enough, the very novelty of the most novel ideas sometimes makes them seem at first to be more derivative than they are. New discoveries often have to be conceived initially as variations of existing things, *even by their discoverers*, because there isn't yet the conceptual vocabulary to express them.

There are definitely some dangers to copying, though. One is that you'll tend to copy old things -- things that were in their day at the frontier of knowledge, but no longer are.

And when you do copy something, don't copy every feature of it. Some will make you ridiculous if you do. Don't copy the manner of an eminent 50 year old professor if you're 18, for example, or the idiom of a Renaissance poem hundreds of years later.

Some of the features of things you admire are flaws they succeeded despite. Indeed, the features that are easiest to imitate are the most likely to be the flaws.

This is particularly true for behavior. Some talented people are jerks, and this sometimes makes it seem to the inexperienced that being a jerk is part of being talented. It isn't; being talented is merely how they get away with it.

One of the most powerful kinds of copying is to copy something from one field into another. History is so full of chance discoveries of this type that it's probably worth giving chance a hand by deliberately learning about other kinds of work. You can take ideas from quite distant fields if you let them be metaphors.

Negative examples can be as inspiring as positive ones. In fact you can sometimes learn more from things done badly than from things done well; sometimes it only becomes clear what's needed when it's missing.

If a lot of the best people in your field are collected in one place, it's usually a good idea to visit for a while. It will increase your ambition, and also, by showing you that these people are human, increase your self- confidence. 26

If you're earnest you'll probably get a warmer welcome than you might expect. Most people who are very good at something are happy to talk about it with anyone who's genuinely interested. If they're really good at their work, then they probably have a hobbyist's interest in it, and hobbyists always want to talk about their hobbies.

It may take some effort to find the people who are really good, though. Doing great work has such prestige that in some places, particularly universities, there's a polite fiction that everyone is engaged in it. And that is far from true. People within universities can't say so openly, but the quality of the work being done in different departments varies immensely. Some departments have people doing great work; others have in the past; others never have.

Seek out the best colleagues. There are a lot of projects that can't be done alone, and even if you're working on one that can be, it's good to have other people to encourage you and to bounce ideas off.

Colleagues don't just affect your work, though; they also affect you. So work with people you want to become like, because you will.

Quality is more important than quantity in colleagues. It's better to have one or two great ones than a building full of pretty good ones. In fact it's not merely better, but necessary, judging from history: the degree to which great work happens in clusters suggests that one's colleagues often make the difference between doing great work and not.

How do you know when you have sufficiently good colleagues? In my experience, when you do, you know. Which means if you're unsure, you probably don't. But it may be possible to give a more concrete answer than that. Here's an attempt: sufficiently good colleagues offer *surprising* insights. They can see and do things that you can't. So if you have a handful of colleagues good enough to keep you on your toes in this sense, you're probably over the threshold.

Most of us can benefit from collaborating with colleagues, but some projects require people on a larger scale, and starting one of those is not for everyone. If you want to run a project like that, you'll have to become a manager, and managing well takes aptitude and interest like any other kind of work. If you don't have them, there is no middle path: you must either force yourself to learn management as a second language, or avoid such projects. 27

Husband your morale. It's the basis of everything when you're working on ambitious projects. You have to nurture and protect it like a living organism.

Morale starts with your view of life. You're more likely to do great work if you're an optimist, and more likely to if you think of yourself as lucky than if you think of yourself as a victim.

Indeed, work can to some extent protect you from your problems. If you choose work that's pure, its very difficulties will serve as a refuge from the difficulties of everyday life. If this is escapism, it's a very productive form of it, and one that has been used by some of the greatest minds in history.

Morale compounds via work: high morale helps you do good work, which increases your morale and helps you do even better work. But this cycle also operates in the other direction: if you're not doing good work, that can demoralize you and make it even harder to. Since it matters so much for this cycle to be running in the right direction, it can be a good idea to switch to easier work when you're stuck, just so you start to get something done.

One of the biggest mistakes ambitious people make is to allow setbacks to destroy their morale all at once, like a balloon bursting. You can inoculate yourself against this by explicitly considering setbacks a part of your process. Solving hard problems always involves some backtracking.

Doing great work is a depth-first search whose root node is the desire to. So "If at first you don't succeed, try, try again" isn't quite right. It should be: If at first you don't succeed, either try again, or backtrack and then try again.

"Never give up" is also not quite right. Obviously there are times when it's the right choice to eject. A more precise version would be: Never let setbacks panic you into backtracking more than you need to. Corollary: Never abandon the root node.

It's not necessarily a bad sign if work is a struggle, any more than it's a bad sign to be out of breath while running. It depends how fast you're running. So learn to distinguish good pain from bad. Good pain is a sign of effort; bad pain is a sign of damage.

An audience is a critical component of morale. If you're a scholar, your audience may be your peers; in the arts, it may be an audience in the traditional sense. Either way it doesn't need to be big. The value of an audience doesn't grow anything like linearly with its size. Which is bad news if you're famous, but good news if you're just starting out, because it means a small but dedicated audience can be enough to sustain you. If a handful of people genuinely love what you're doing, that's enough.

To the extent you can, avoid letting intermediaries come between you and your audience. In some types of work this is inevitable, but it's so liberating to escape it that you might be better off switching to an adjacent type if that will let you go direct. 28

The people you spend time with will also have a big effect on your morale. You'll find there are some who increase your energy and others who decrease it, and the effect someone has is not always what you'd expect. Seek out the people who increase your energy and avoid those who decrease it. Though of course if there's someone you need to take care of, that takes precedence.

Don't marry someone who doesn't understand that you need to work, or sees your work as competition for your attention. If you're ambitious, you need to work; it's almost like a medical condition; so someone who won't let you work either doesn't understand you, or does and doesn't care.

Ultimately morale is physical. You think with your body, so it's important to take care of it. That means exercising regularly, eating and sleeping well, and avoiding the more dangerous kinds of drugs. Running and walking are particularly good forms of exercise because they're good for thinking. 29

People who do great work are not necessarily happier than everyone else, but they're happier than they'd be if they didn't. In fact, if you're smart and ambitious, it's dangerous *not* to be productive. People who are smart and ambitious but don't achieve much tend to become bitter.

It's ok to want to impress other people, but choose the right people. The opinion of people you respect is signal. Fame, which is the opinion of a much larger group you might or might not respect, just adds noise.

The prestige of a type of work is at best a trailing indicator and sometimes completely mistaken. If you do anything well enough, you'll make it prestigious.

So the question to ask about a type of work is not how much prestige it has, but how well it could be done.

Competition can be an effective motivator, but don't let it choose the problem for you; don't let yourself get drawn into chasing something just because others are. In fact, don't let competitors make you do anything much more specific than work harder.

Curiosity is the best guide. Your curiosity never lies, and it knows more than you do about what's worth paying attention to.

Notice how often that word has come up. If you asked an oracle the secret to doing great work and the oracle replied with a single word, my bet would be on "curiosity."

That doesn't translate directly to advice. It's not enough just to be curious, and you can't command curiosity anyway. But you can nurture it and let it drive you.

Curiosity is the key to all four steps in doing great work: it will choose the field for you, get you to the frontier, cause you to notice the gaps in it, and drive you to explore them. The whole process is a kind of dance with curiosity.

Believe it or not, I tried to make this essay as short as I could. But its length at least means it acts as a filter. If you made it this far, you must be interested in doing great work. And if so you're already further along than you might realize, because the set of people willing to want to is small.

The factors in doing great work are factors in the literal, mathematical sense, and they are: ability, interest, effort, and luck. Luck by definition you can't do anything about, so we can ignore that. And we can assume effort, if you do in fact want to do great work. So the problem boils down to ability and interest. Can you find a kind of work where your ability and interest will combine to yield an explosion of new ideas?

Here there are grounds for optimism. There are so many different ways to do great work, and even more that are still undiscovered. Out of all those different types of work, the one you're most suited for is probably a pretty close match. Probably a comically close match. It's just a question of finding it, and how far into it your ability and interest can take you. And you can only answer that by trying.

Many more people could try to do great work than do. What holds them back is a combination of modesty and fear. It seems presumptuous to try to be Newton or Shakespeare. It also seems hard; surely if you tried something like that, you'd fail. Presumably the calculation is rarely explicit. Few people consciously decide not to try to do great work. But that's what's going on subconsciously; they shy away from the question.

So I'm going to pull a sneaky trick on you. Do you want to do great work, or not? Now you have to decide consciously. Sorry about that. I wouldn't have

done it to a general audience. But we already know you're interested.

Don't worry about being presumptuous. You don't have to tell anyone. And if it's too hard and you fail, so what? Lots of people have worse problems than that. In fact you'll be lucky if it's the worst problem you have.

Yes, you'll have to work hard. But again, lots of people have to work hard. And if you're working on something you find very interesting, which you necessarily will if you're on the right path, the work will probably feel less burdensome than a lot of your peers'.

The discoveries are out there, waiting to be made. Why not by you?

Notes

[1] I don't think you could give a precise definition of what counts as great work. Doing great work means doing something important so well that you expand people's ideas of what's possible. But there's no threshold for importance. It's a matter of degree, and often hard to judge at the time anyway. So I'd rather people focused on developing their interests rather than worrying about whether they're important or not. Just try to do something amazing, and leave it to future generations to say if you succeeded.

[2] A lot of standup comedy is based on noticing anomalies in everyday life. "Did you ever notice...?" New ideas come from doing this about nontrivial things. Which may help explain why people's reaction to a new idea is often the first half of laughing: Ha!

[3] That second qualifier is critical. If you're excited about something most authorities discount, but you can't give a more precise explanation than "they don't get it," then you're starting to drift into the territory of cranks.

[4] Finding something to work on is not simply a matter of finding a match between the current version of you and a list of known problems. You'll often have to coevolve with the problem. That's why it can sometimes be so hard to figure out what to work on. The search space is huge. It's the cartesian product of all possible types of work, both known and yet to be discovered, and all possible future versions of you.

There's no way you could search this whole space, so you have to rely on heuristics to generate promising paths through it and hope the best matches will be clustered. Which they will not always be; different types of work have been collected together as much by accidents of history as by the intrinsic similarities between them.

[5] There are many reasons curious people are more likely to do great work, but one of the more subtle is that, by casting a wide net, they're more likely to find the right thing to work on in the first place.

[6] It can also be dangerous to make things for an audience you feel is less sophisticated than you, if that causes you to talk down to them. You can make

a lot of money doing that, if you do it in a sufficiently cynical way, but it's not the route to great work. Not that anyone using this m.o. would care.

[7] This idea I learned from Hardy's *A Mathematician's Apology*, which I recommend to anyone ambitious to do great work, in any field.

[8] Just as we overestimate what we can do in a day and underestimate what we can do over several years, we overestimate the damage done by procrastinating for a day and underestimate the damage done by procrastinating for several years.

[9] You can't usually get paid for doing exactly what you want, especially early on. There are two options: get paid for doing work close to what you want and hope to push it closer, or get paid for doing something else entirely and do your own projects on the side. Both can work, but both have drawbacks: in the first approach your work is compromised by default, and in the second you have to fight to get time to do it.

[10] If you set your life up right, it will deliver the focus-relax cycle automatically. The perfect setup is an office you work in and that you walk to and from.

[11] There may be some very unworldly people who do great work without consciously trying to. If you want to expand this rule to cover that case, it becomes: Don't try to be anything except the best.

[12] This gets more complicated in work like acting, where the goal is to adopt a fake persona. But even here it's possible to be affected. Perhaps the rule in such fields should be to avoid *unintentional* affectation.

[13] It's safe to have beliefs that you treat as unquestionable if and only if they're also unfalsifiable. For example, it's safe to have the principle that everyone should be treated equally under the law, because a sentence with a "should" in it isn't really a statement about the world and is therefore hard to disprove. And if there's no evidence that could disprove one of your principles, there can't be any facts you'd need to ignore in order to preserve it.

[14] Affectation is easier to cure than intellectual dishonesty. Affectation is often a shortcoming of the young that burns off in time, while intellectual dishonesty is more of a character flaw.

[15] Obviously you don't have to be working at the exact moment you have the idea, but you'll probably have been working fairly recently.

[16] Some say psychoactive drugs have a similar effect. I'm skeptical, but also almost totally ignorant of their effects.

[17] For example you might give the n th most important topic $(m-1)/m^n$ of your attention, for some $m > 1$. You couldn't allocate your attention so precisely, of course, but this at least gives an idea of a reasonable distribution.

[18] The principles defining a religion have to be mistaken. Otherwise anyone might adopt them, and there would be nothing to distinguish the adherents of

the religion from everyone else.

[19] It might be a good exercise to try writing down a list of questions you wondered about in your youth. You might find you're now in a position to do something about some of them.

[20] The connection between originality and uncertainty causes a strange phenomenon: because the conventional-minded are more certain than the independent-minded, this tends to give them the upper hand in disputes, even though they're generally stupider.

The best lack all conviction, while the worst > Are full of passionate intensity.

[21] Derived from Linus Pauling's "If you want to have good ideas, you must have many ideas."

[22] Attacking a project as a "toy" is similar to attacking a statement as "inappropriate." It means that no more substantial criticism can be made to stick.

[23] One way to tell whether you're wasting time is to ask if you're producing or consuming. Writing computer games is less likely to be a waste of time than playing them, and playing games where you create something is less likely to be a waste of time than playing games where you don't.

[24] Another related advantage is that if you haven't said anything publicly yet, you won't be biased toward evidence that supports your earlier conclusions. With sufficient integrity you could achieve eternal youth in this respect, but few manage to. For most people, having previously published opinions has an effect similar to ideology, just in quantity 1.

[25] In the early 1630s Daniel Mytens made a painting of Henrietta Maria handing a laurel wreath to Charles I. Van Dyck then painted his own version to show how much better he was.

[26] I'm being deliberately vague about what a place is. As of this writing, being in the same physical place has advantages that are hard to duplicate, but that could change.

[27] This is false when the work the other people have to do is very constrained, as with SETI@home or Bitcoin. It may be possible to expand the area in which it's false by defining similarly restricted protocols with more freedom of action in the nodes.

[28] Corollary: Building something that enables people to go around intermediaries and engage directly with their audience is probably a good idea.

[29] It may be helpful always to walk or run the same route, because that frees attention for thinking. It feels that way to me, and there is some historical evidence for it.

Thanks to Trevor Blackwell, Daniel Gackle, Pam Graham, Tom Howard, Patrick Hsu, Steve Huffman, Jessica Livingston, Henry Lloyd-Baker, Bob Metcalfe, Ben Miller, Robert Morris, Michael Nielsen, Courtenay Pipkin, Joris Poort, Mieke Roos, Rajat Suri, Harj Taggar, Garry Tan, and my younger son for suggestions and for reading drafts.

219 Superlinear Returns

October 2023

One of the most important things I didn't understand about the world when I was a child is the degree to which the returns for performance are superlinear.

Teachers and coaches implicitly told us the returns were linear. "You get out," I heard a thousand times, "what you put in." They meant well, but this is rarely true. If your product is only half as good as your competitor's, you don't get half as many customers. You get no customers, and you go out of business.

It's obviously true that the returns for performance are superlinear in business. Some think this is a flaw of capitalism, and that if we changed the rules it would stop being true. But superlinear returns for performance are a feature of the world, not an artifact of rules we've invented. We see the same pattern in fame, power, military victories, knowledge, and even benefit to humanity. In all of these, the rich get richer. ¹

You can't understand the world without understanding the concept of superlinear returns. And if you're ambitious you definitely should, because this will be the wave you surf on.

It may seem as if there are a lot of different situations with superlinear returns, but as far as I can tell they reduce to two fundamental causes: exponential growth and thresholds.

The most obvious case of superlinear returns is when you're working on something that grows exponentially. For example, growing bacterial cultures. When they grow at all, they grow exponentially. But they're tricky to grow. Which means the difference in outcome between someone who's adept at it and someone who's not is very great.

Startups can also grow exponentially, and we see the same pattern there. Some manage to achieve high growth rates. Most don't. And as a result you get qualitatively different outcomes: the companies with high growth rates tend to become immensely valuable, while the ones with lower growth rates may not even survive.

Y Combinator encourages founders to focus on growth rate rather than absolute numbers. It prevents them from being discouraged early on, when the absolute

numbers are still low. It also helps them decide what to focus on: you can use growth rate as a compass to tell you how to evolve the company. But the main advantage is that by focusing on growth rate you tend to get something that grows exponentially.

YC doesn't explicitly tell founders that with growth rate "you get out what you put in," but it's not far from the truth. And if growth rate were proportional to performance, then the reward for performance p over time t would be proportional to $p t$.

Even after decades of thinking about this, I find that sentence startling.

Whenever how well you do depends on how well you've done, you'll get exponential growth. But neither our DNA nor our customs prepare us for it. No one finds exponential growth natural; every child is surprised, the first time they hear it, by the story of the man who asks the king for a single grain of rice the first day and double the amount each successive day.

What we don't understand naturally we develop customs to deal with, but we don't have many customs about exponential growth either, because there have been so few instances of it in human history. In principle herding should have been one: the more animals you had, the more offspring they'd have. But in practice grazing land was the limiting factor, and there was no plan for growing that exponentially.

Or more precisely, no generally applicable plan. There *was* a way to grow one's territory exponentially: by conquest. The more territory you control, the more powerful your army becomes, and the easier it is to conquer new territory. This is why history is full of empires. But so few people created or ran empires that their experiences didn't affect customs very much. The emperor was a remote and terrifying figure, not a source of lessons one could use in one's own life.

The most common case of exponential growth in preindustrial times was probably scholarship. The more you know, the easier it is to learn new things. The result, then as now, was that some people were startlingly more knowledgeable than the rest about certain topics. But this didn't affect customs much either. Although empires of ideas can overlap and there can thus be far more emperors, in preindustrial times this type of empire had little practical effect. 2

That has changed in the last few centuries. Now the emperors of ideas can design bombs that defeat the emperors of territory. But this phenomenon is still so new that we haven't fully assimilated it. Few even of the participants realize they're benefitting from exponential growth or ask what they can learn from other instances of it.

The other source of superlinear returns is embodied in the expression "winner take all." In a sports match the relationship between performance and return is a step function: the winning team gets one win whether they do much better or just slightly better. 3

The source of the step function is not competition per se, however. It's that there are thresholds in the outcome. You don't need competition to get those. There can be thresholds in situations where you're the only participant, like proving a theorem or hitting a target.

It's remarkable how often a situation with one source of superlinear returns also has the other. Crossing thresholds leads to exponential growth: the winning side in a battle usually suffers less damage, which makes them more likely to win in the future. And exponential growth helps you cross thresholds: in a market with network effects, a company that grows fast enough can shut out potential competitors.

Fame is an interesting example of a phenomenon that combines both sources of superlinear returns. Fame grows exponentially because existing fans bring you new ones. But the fundamental reason it's so concentrated is thresholds: there's only so much room on the A-list in the average person's head.

The most important case combining both sources of superlinear returns may be learning. Knowledge grows exponentially, but there are also thresholds in it. Learning to ride a bicycle, for example. Some of these thresholds are akin to machine tools: once you learn to read, you're able to learn anything else much faster. But the most important thresholds of all are those representing new discoveries. Knowledge seems to be fractal in the sense that if you push hard at the boundary of one area of knowledge, you sometimes discover a whole new field. And if you do, you get first crack at all the new discoveries to be made in it. Newton did this, and so did Durer and Darwin.

Are there general rules for finding situations with superlinear returns? The most obvious one is to seek work that compounds.

There are two ways work can compound. It can compound directly, in the sense that doing well in one cycle causes you to do better in the next. That happens for example when you're building infrastructure, or growing an audience or brand. Or work can compound by teaching you, since learning compounds. This second case is an interesting one because you may feel you're doing badly as it's happening. You may be failing to achieve your immediate goal. But if you're learning a lot, then you're getting exponential growth nonetheless.

This is one reason Silicon Valley is so tolerant of failure. People in Silicon Valley aren't blindly tolerant of failure. They'll only continue to bet on you if you're learning from your failures. But if you are, you are in fact a good bet: maybe your company didn't grow the way you wanted, but you yourself have, and that should yield results eventually.

Indeed, the forms of exponential growth that don't consist of learning are so often intermixed with it that we should probably treat this as the rule rather than the exception. Which yields another heuristic: always be learning. If you're not learning, you're probably not on a path that leads to superlinear returns.

But don't overoptimize *what* you're learning. Don't limit yourself to learning

things that are already known to be valuable. You're learning; you don't know for sure yet what's going to be valuable, and if you're too strict you'll lop off the outliers.

What about step functions? Are there also useful heuristics of the form "seek thresholds" or "seek competition?" Here the situation is trickier. The existence of a threshold doesn't guarantee the game will be worth playing. If you play a round of Russian roulette, you'll be in a situation with a threshold, certainly, but in the best case you're no better off. "Seek competition" is similarly useless; what if the prize isn't worth competing for? Sufficiently fast exponential growth guarantees both the shape and magnitude of the return curve -- because something that grows fast enough will grow big even if it's trivially small at first -- but thresholds only guarantee the shape. 4

A principle for taking advantage of thresholds has to include a test to ensure the game is worth playing. Here's one that does: if you come across something that's mediocre yet still popular, it could be a good idea to replace it. For example, if a company makes a product that people dislike yet still buy, then presumably they'd buy a better alternative if you made one. 5

It would be great if there were a way to find promising intellectual thresholds. Is there a way to tell which questions have whole new fields beyond them? I doubt we could ever predict this with certainty, but the prize is so valuable that it would be useful to have predictors that were even a little better than random, and there's hope of finding those. We can to some degree predict when a research problem *isn't* likely to lead to new discoveries: when it seems legit but boring. Whereas the kind that do lead to new discoveries tend to seem very mystifying, but perhaps unimportant. (If they were mystifying and obviously important, they'd be famous open questions with lots of people already working on them.) So one heuristic here is to be driven by curiosity rather than careerism -- to give free rein to your curiosity instead of working on what you're supposed to.

The prospect of superlinear returns for performance is an exciting one for the ambitious. And there's good news in this department: this territory is expanding in both directions. There are more types of work in which you can get superlinear returns, and the returns themselves are growing.

There are two reasons for this, though they're so closely intertwined that they're more like one and a half: progress in technology, and the decreasing importance of organizations.

Fifty years ago it used to be much more necessary to be part of an organization to work on ambitious projects. It was the only way to get the resources you needed, the only way to have colleagues, and the only way to get distribution. So in 1970 your prestige was in most cases the prestige of the organization you belonged to. And prestige was an accurate predictor, because if you weren't part of an organization, you weren't likely to achieve much. There were a handful of exceptions, most notably artists and writers, who worked alone using inexpensive

tools and had their own brands. But even they were at the mercy of organizations for reaching audiences. 6

A world dominated by organizations damped variation in the returns for performance. But this world has eroded significantly just in my lifetime. Now a lot more people can have the freedom that artists and writers had in the 20th century. There are lots of ambitious projects that don't require much initial funding, and lots of new ways to learn, make money, find colleagues, and reach audiences.

There's still plenty of the old world left, but the rate of change has been dramatic by historical standards. Especially considering what's at stake. It's hard to imagine a more fundamental change than one in the returns for performance.

Without the damping effect of institutions, there will be more variation in outcomes. Which doesn't imply everyone will be better off: people who do well will do even better, but those who do badly will do worse. That's an important point to bear in mind. Exposing oneself to superlinear returns is not for everyone. Most people will be better off as part of the pool. So who should shoot for superlinear returns? Ambitious people of two types: those who know they're so good that they'll be net ahead in a world with higher variation, and those, particularly the young, who can afford to risk trying it to find out. 7

The switch away from institutions won't simply be an exodus of their current inhabitants. Many of the new winners will be people they'd never have let in. So the resulting democratization of opportunity will be both greater and more authentic than any tame intramural version the institutions themselves might have cooked up.

Not everyone is happy about this great unlocking of ambition. It threatens some vested interests and contradicts some ideologies. 8 But if you're an ambitious individual it's good news for you. How should you take advantage of it?

The most obvious way to take advantage of superlinear returns for performance is by doing exceptionally good work. At the far end of the curve, incremental effort is a bargain. All the more so because there's less competition at the far end -- and not just for the obvious reason that it's hard to do something exceptionally well, but also because people find the prospect so intimidating that few even try. Which means it's not just a bargain to do exceptional work, but a bargain even to try to.

There are many variables that affect how good your work is, and if you want to be an outlier you need to get nearly all of them right. For example, to do something exceptionally well, you have to be interested in it. Mere diligence is not enough. So in a world with superlinear returns, it's even more valuable to know what you're interested in, and to find ways to work on it. 9 It will also be important to choose work that suits your circumstances. For example, if there's a kind of work that inherently requires a huge expenditure of time and energy, it will be increasingly valuable to do it when you're young and don't yet have

children.

There's a surprising amount of technique to doing great work. It's not just a matter of trying hard. I'm going to take a shot giving a recipe in one paragraph.

Choose work you have a natural aptitude for and a deep interest in. Develop a habit of working on your own projects; it doesn't matter what they are so long as you find them excitingly ambitious. Work as hard as you can without burning out, and this will eventually bring you to one of the frontiers of knowledge. These look smooth from a distance, but up close they're full of gaps. Notice and explore such gaps, and if you're lucky one will expand into a whole new field. Take as much risk as you can afford; if you're not failing occasionally you're probably being too conservative. Seek out the best colleagues. Develop good taste and learn from the best examples. Be honest, especially with yourself. Exercise and eat and sleep well and avoid the more dangerous drugs. When in doubt, follow your curiosity. It never lies, and it knows more than you do about what's worth paying attention to. 10

And there is of course one other thing you need: to be lucky. Luck is always a factor, but it's even more of a factor when you're working on your own rather than as part of an organization. And though there are some valid aphorisms about luck being where preparedness meets opportunity and so on, there's also a component of true chance that you can't do anything about. The solution is to take multiple shots. Which is another reason to start taking risks early.

The best example of a field with superlinear returns is probably science. It has exponential growth, in the form of learning, combined with thresholds at the extreme edge of performance -- literally at the limits of knowledge.

The result has been a level of inequality in scientific discovery that makes the wealth inequality of even the most stratified societies seem mild by comparison. Newton's discoveries were arguably greater than all his contemporaries' combined. 11

This point may seem obvious, but it might be just as well to spell it out. Superlinear returns imply inequality. The steeper the return curve, the greater the variation in outcomes.

In fact, the correlation between superlinear returns and inequality is so strong that it yields another heuristic for finding work of this type: look for fields where a few big winners outperform everyone else. A kind of work where everyone does about the same is unlikely to be one with superlinear returns.

What are fields where a few big winners outperform everyone else? Here are some obvious ones: sports, politics, art, music, acting, directing, writing, math, science, starting companies, and investing. In sports the phenomenon is due to externally imposed thresholds; you only need to be a few percent faster to win every race. In politics, power grows much as it did in the days of emperors. And in some of the other fields (including politics) success is driven largely by fame, which has its own source of superlinear growth. But when we exclude sports and

politics and the effects of fame, a remarkable pattern emerges: the remaining list is exactly the same as the list of fields where you have to be *independent-minded* to succeed -- where your ideas have to be not just correct, but novel as well. 12

This is obviously the case in science. You can't publish papers saying things that other people have already said. But it's just as true in investing, for example. It's only useful to believe that a company will do well if most other investors don't; if everyone else thinks the company will do well, then its stock price will already reflect that, and there's no room to make money.

What else can we learn from these fields? In all of them you have to put in the initial effort. Superlinear returns seem small at first. *At this rate*, you find yourself thinking, *I'll never get anywhere*. But because the reward curve rises so steeply at the far end, it's worth taking extraordinary measures to get there.

In the startup world, the name for this principle is "do things that don't scale." If you pay a ridiculous amount of attention to your tiny initial set of customers, ideally you'll kick off exponential growth by word of mouth. But this same principle applies to anything that grows exponentially. Learning, for example. When you first start learning something, you feel lost. But it's worth making the initial effort to get a toehold, because the more you learn, the easier it will get.

There's another more subtle lesson in the list of fields with superlinear returns: not to equate work with a job. For most of the 20th century the two were identical for nearly everyone, and as a result we've inherited a custom that equates productivity with having a job. Even now to most people the phrase "your work" means their job. But to a writer or artist or scientist it means whatever they're currently studying or creating. For someone like that, their work is something they carry with them from job to job, if they have jobs at all. It may be done for an employer, but it's part of their portfolio.

It's an intimidating prospect to enter a field where a few big winners outperform everyone else. Some people do this deliberately, but you don't need to. If you have sufficient natural ability and you follow your curiosity sufficiently far, you'll end up in one. Your curiosity won't let you be interested in boring questions, and interesting questions tend to create fields with superlinear returns if they're not already part of one.

The territory of superlinear returns is by no means static. Indeed, the most extreme returns come from expanding it. So while both ambition and curiosity can get you into this territory, curiosity may be the more powerful of the two. Ambition tends to make you climb existing peaks, but if you stick close enough to an interesting enough question, it may grow into a mountain beneath you.

Notes

There's a limit to how sharply you can distinguish between effort, performance, and return, because they're not sharply distinguished in fact. What counts as return to one person might be performance to another. But though the borders

of these concepts are blurry, they're not meaningless. I've tried to write about them as precisely as I could without crossing into error.

[1] Evolution itself is probably the most pervasive example of superlinear returns for performance. But this is hard for us to empathize with because we're not the recipients; we're the returns.

[2] Knowledge did of course have a practical effect before the Industrial Revolution. The development of agriculture changed human life completely. But this kind of change was the result of broad, gradual improvements in technique, not the discoveries of a few exceptionally learned people.

[3] It's not mathematically correct to describe a step function as superlinear, but a step function starting from zero works like a superlinear function when it describes the reward curve for effort by a rational actor. If it starts at zero then the part before the step is below any linearly increasing return, and the part after the step must be above the necessary return at that point or no one would bother.

[4] Seeking competition could be a good heuristic in the sense that some people find it motivating. It's also somewhat of a guide to promising problems, because it's a sign that other people find them promising. But it's a very imperfect sign: often there's a clamoring crowd chasing some problem, and they all end up being trumped by someone quietly working on another one.

[5] Not always, though. You have to be careful with this rule. When something is popular despite being mediocre, there's often a hidden reason why. Perhaps monopoly or regulation make it hard to compete. Perhaps customers have bad taste or have broken procedures for deciding what to buy. There are huge swathes of mediocre things that exist for such reasons.

[6] In my twenties I wanted to be an *artist* and even went to art school to study painting. Mostly because I liked art, but a nontrivial part of my motivation came from the fact that artists seemed least at the mercy of organizations.

[7] In principle everyone is getting superlinear returns. Learning compounds, and everyone learns in the course of their life. But in practice few push this kind of everyday learning to the point where the return curve gets really steep.

[8] It's unclear exactly what advocates of "equity" mean by it. They seem to disagree among themselves. But whatever they mean is probably at odds with a world in which institutions have less power to control outcomes, and a handful of outliers do much better than everyone else.

It may seem like bad luck for this concept that it arose at just the moment when the world was shifting in the opposite direction, but I don't think this was a coincidence. I think one reason it arose now is because its adherents feel threatened by rapidly increasing variation in performance.

[9] Corollary: Parents who pressure their kids to work on something prestigious, like medicine, even though they have no interest in it, will be hosing them even

more than they have in the past.

[10] The original version of this paragraph was the first draft of "*How to Do Great Work*." As soon as I wrote it I realized it was a more important topic than superlinear returns, so I paused the present essay to expand this paragraph into its own. Practically nothing remains of the original version, because after I finished "How to Do Great Work" I rewrote it based on that.

[11] Before the Industrial Revolution, people who got rich usually did it like emperors: capturing some resource made them more powerful and enabled them to capture more. Now it can be done like a scientist, by discovering or building something uniquely valuable. Most people who get rich use a mix of the old and the new ways, but in the most advanced economies the ratio has *shifted dramatically* toward discovery just in the last half century.

[12] It's not surprising that conventional-minded people would dislike inequality if independent-mindedness is one of the biggest drivers of it. But it's not simply that they don't want anyone to have what they can't. The conventional-minded literally can't imagine what it's like to have novel ideas. So the whole phenomenon of great variation in performance seems unnatural to them, and when they encounter it they assume it must be due to cheating or to some malign external influence.

Thanks to Trevor Blackwell, Patrick Collison, Tyler Cowen, Jessica Livingston, Harj Taggar, and Garry Tan for reading drafts of this.
