How to Start a Startup

March 2005(This essay is derived from a talk at the Harvard Computer  
Society.)You need three things to create a successful startup: to start with  
good people, to make something customers actually want, and to spend  
as little money as possible. Most startups that fail do it because  
they fail at one of these. A startup that does all three will  
probably succeed.And that's kind of exciting, when you think about it, because all  
three are doable. Hard, but doable. And since a startup that  
succeeds ordinarily makes its founders rich, that implies getting  
rich is doable too. Hard, but doable.If there is one message I'd like to get across about startups,  
that's it. There is no magically difficult step that requires  
brilliance to solve.The IdeaIn particular, you don't need a brilliant   
idea to start a startup  
around. The way a startup makes money is to offer people better  
technology than they have now. But what people have now is often  
so bad that it doesn't take brilliance to do better.Google's plan, for example, was simply to create a search site that  
didn't suck. They had three new ideas: index more of the Web, use  
links to rank search results, and have clean, simple web pages with  
unintrusive keyword-based ads. Above all, they were determined to  
make a site that was good to use. No doubt there are great technical  
tricks within Google, but the overall plan was straightforward.  
And while they probably have bigger ambitions now, this alone brings  
them a billion dollars a year. [1]There are plenty of other areas that are just as backward as search  
was before Google. I can think of several heuristics for generating  
ideas for startups, but most reduce to this: look at something  
people are trying to do, and figure out how to do it in a way that  
doesn't suck.For example, dating sites currently suck far worse than search did  
before Google. They all use the same simple-minded model.  
They seem to have approached the problem by thinking about how to  
do database matches instead of how dating works in the real world.  
An undergrad could build something better as a class project. And  
yet there's a lot of money at stake. Online dating is a valuable  
business now, and it might be worth a hundred times as much if it  
worked.An idea for a startup, however, is only a beginning. A lot of  
would-be startup founders think the key to the whole process is the  
initial idea, and from that point all you have to do is execute.  
Venture capitalists know better. If you go to VC firms with a  
brilliant idea that you'll tell them about if they sign a nondisclosure  
agreement, most will tell you to get lost. That shows how much a   
mere idea is worth. The market price is less than the inconvenience   
of signing an NDA.Another sign of how little the initial idea is worth is the number  
of startups that change their plan en route. Microsoft's original  
plan was to make money selling programming languages, of all things.  
Their current business model didn't occur to them until IBM dropped  
it in their lap five years later.Ideas for startups are worth something, certainly, but the trouble  
is, they're not transferrable. They're not something you could  
hand to someone else to execute. Their value is mainly as starting  
points: as questions for the people who had them to continue thinking  
about.What matters is not ideas, but the people who have them. Good  
people can fix bad ideas, but good ideas can't save bad people.   
PeopleWhat do I mean by good people? One of the best tricks I learned   
during our startup was a rule for deciding   
who to hire. Could you  
describe the person as an animal? It might be hard to translate   
that into another language, but I think everyone in the US knows   
what it means. It means someone who takes their work a little too   
seriously; someone who does what they do so well that they pass  
right through professional and cross over into obsessive.What it means specifically depends on the job: a salesperson who  
just won't take no for an answer; a hacker who will stay up till   
4:00 AM rather than go to bed leaving code with a bug in it; a PR   
person who will cold-call New York Times reporters on their cell  
phones; a graphic designer who feels physical pain when something   
is two millimeters out of place.Almost everyone who worked for us was an animal at what they did.   
The woman in charge of sales was so tenacious that I used to feel  
sorry for potential customers on the phone with her. You could   
sense them squirming on the hook, but you knew there would be no   
rest for them till they'd signed up.If you think about people you know, you'll find the animal test is  
easy to apply. Call the person's image to mind and imagine the  
sentence "so-and-so is an animal." If you laugh, they're not. You  
don't need or perhaps even want this quality in big companies, but  
you need it in a startup.For programmers we had three additional tests. Was the person  
genuinely smart? If so, could they actually get things done? And  
finally, since a few good hackers have unbearable personalities,   
could we stand to have them around?That last test filters out surprisingly few people. We could bear  
any amount of nerdiness if someone was truly smart. What we couldn't  
stand were people with a lot of attitude. But most of those weren't  
truly smart, so our third test was largely a restatement of the  
first.When nerds are unbearable it's usually because they're trying too  
hard to seem smart. But the smarter they are, the less pressure  
they feel to act smart. So as a rule you can recognize genuinely  
smart people by their ability to say things like "I don't know,"   
"Maybe you're right," and "I don't understand x well enough."This technique doesn't always work, because people can be influenced  
by their environment. In the MIT CS department, there seems to be  
a tradition of acting like a brusque know-it-all. I'm told it derives  
ultimately from Marvin Minsky, in the same way the classic airline  
pilot manner is said to derive from Chuck Yeager. Even genuinely  
smart people start to act this way there, so you have to make  
allowances.It helped us to have Robert Morris, who is one of the readiest to  
say "I don't know" of anyone I've met. (At least, he was before he   
became a professor at MIT.) No one dared put on attitude around   
Robert, because he was obviously smarter than they were and yet had  
zero attitude himself.Like most startups, ours began with a group of friends, and it was  
through personal contacts that we got most of the people we hired.  
This is a crucial difference between startups and big companies.  
Being friends with someone for even a couple days will tell you   
more than companies could ever learn in interviews. [2]It's no coincidence that startups start around universities, because  
that's where smart people meet. It's not what people learn in   
classes at MIT and Stanford that has made technology companies  
spring up around them. They could sing campfire songs in the classes  
so long as admissions worked the same.If you start a startup, there's a good chance it will be with people  
you know from college or grad school. So in theory you ought to   
try to make friends with as many smart people as you can in school,  
right? Well, no. Don't make a conscious effort to schmooze; that  
doesn't work well with hackers.What you should do in college is work on your own projects. Hackers  
should do this even if they don't plan to start startups, because   
it's the only real way to learn how to program. In some cases you  
may collaborate with other students, and this is the best way to  
get to know good hackers. The project may even grow into a startup.  
But once again, I wouldn't aim too directly at either target. Don't  
force things; just work on stuff you like with people you like.Ideally you want between two and four founders. It would be hard  
to start with just one. One person would find the moral weight of  
starting a company hard to bear. Even Bill Gates, who seems to be   
able to bear a good deal of moral weight, had to have a co-founder.   
But you don't want so many founders that the company starts to look  
like a group photo. Partly because you don't need a lot of people  
at first, but mainly because the more founders you have, the worse  
disagreements you'll have. When there are just two or three founders,  
you know you have to resolve disputes immediately or perish. If  
there are seven or eight, disagreements can linger and harden into  
factions. You don't want mere voting; you need unanimity.In a technology startup, which most startups are, the founders  
should include technical people. During the Internet Bubble there   
were a number of startups founded by business people who then went  
looking for hackers to create their product for them. This doesn't   
work well. Business people are bad at deciding what to do with   
technology, because they don't know what the options are, or which  
kinds of problems are hard and which are easy. And when business  
people try to hire hackers, they can't tell which ones are   
good.  
Even other hackers have a hard time doing that.   
For business people it's roulette.Do the founders of a startup have to include business people? That  
depends. We thought so when we started ours, and we asked several   
people who were said to know about this mysterious thing called  
"business" if they would be the president. But they all said no,  
so I had to do it myself. And what I discovered was that business  
was no great mystery. It's not something like physics or medicine  
that requires extensive study. You just try to get people to pay  
you for stuff.I think the reason I made such a mystery of business was that I was  
disgusted by the idea of doing it. I wanted to work in the pure,   
intellectual world of software, not deal with customers' mundane   
problems. People who don't want to get dragged into some kind of  
work often develop a protective incompetence at it. Paul Erdos was  
particularly good at this. By seeming unable even to cut a grapefruit  
in half (let alone go to the store and buy one), he forced other  
people to do such things for him, leaving all his time free for  
math. Erdos was an extreme case, but most husbands use the same   
trick to some degree.Once I was forced to discard my protective incompetence, I found  
that business was neither so hard nor so boring as I feared. There  
are esoteric areas of business that are quite hard, like tax law  
or the pricing of derivatives, but you don't need to know about   
those in a startup. All you need to know about business to run a   
startup are commonsense things people knew before there were business  
schools, or even universities.If you work your way down the Forbes 400 making an x next to the   
name of each person with an MBA, you'll learn something important  
about business school. After Warren Buffett, you don't hit another   
MBA till number 22,  
Phil Knight, the CEO of Nike. There are only 5 MBAs in the top  
50. What you notice in the Forbes 400 are a lot of people with   
technical backgrounds. Bill Gates, Steve Jobs, Larry Ellison,  
Michael Dell, Jeff Bezos, Gordon Moore. The rulers of the technology  
business tend to come from technology, not business. So if you   
want to invest two years in something that will help you succeed   
in business, the evidence suggests you'd do better to learn how to   
hack than get an MBA. [3]There is one reason you might want to include business people in a  
startup, though: because you have to have at least one person willing  
and able to focus on what customers want. Some believe only business  
people can do this-- that hackers can implement software, but not   
design it. That's nonsense. There's nothing about knowing how to  
program that prevents hackers from understanding users, or about  
not knowing how to program that magically enables business people   
to understand them.If you can't understand users, however, you should either learn how  
or find a co-founder who can. That is the single most important  
issue for technology startups, and the rock that sinks more of them  
than anything else.What Customers WantIt's not just startups that have to worry about this. I think most  
businesses that fail do it because they don't give customers what  
they want. Look at restaurants. A large percentage fail, about a  
quarter in the first year. But can you think of one restaurant  
that had really good food and went out of business?Restaurants with great food seem to prosper no matter what. A   
restaurant with great food can be expensive, crowded, noisy, dingy,  
out of the way, and even have bad service, and people will keep  
coming. It's true that a restaurant with mediocre food can sometimes   
attract customers through gimmicks. But that approach is very   
risky. It's more straightforward just to make the food good.It's the same with technology. You hear all kinds of reasons why  
startups fail. But can you think of one that had a massively popular  
product and still failed?In nearly every failed startup, the real problem was that customers  
didn't want the product. For most, the cause of death is listed   
as "ran out of funding," but that's only the immediate cause. Why   
couldn't they get more funding? Probably because the product was  
a dog, or never seemed likely to be done, or both.When I was trying to think of the things every startup needed to   
do, I almost included a fourth: get a version 1 out as soon as you  
can. But I decided not to, because that's implicit in making  
something customers want. The only way to make something customers  
want is to get a prototype in front of them and refine it based on   
their reactions.The other approach is what I call the "Hail Mary" strategy. You   
make elaborate plans for a product, hire a team of engineers to   
develop it (people who do this tend to use the term "engineer" for   
hackers), and then find after a year that you've spent two million  
dollars to develop something no one wants. This was not uncommon  
during the Bubble, especially in companies run by business types,   
who thought of software development as something terrifying that  
therefore had to be carefully planned.We never even considered that approach. As a Lisp hacker, I come   
from the tradition of rapid prototyping. I would not claim (at  
least, not here) that this is the right way to write every program,  
but it's certainly the right way to write software for a startup.  
In a startup, your initial plans are almost certain to be wrong in  
some way, and your first priority should be to figure out where.   
The only way to do that is to try implementing them.Like most startups, we changed our plan on the fly. At first we  
expected our customers to be Web consultants. But it turned out  
they didn't like us, because our software was easy to use and we hosted  
the site. It would be too easy for clients to fire them. We also  
thought we'd be able to sign up a lot of catalog companies, because  
selling online was a natural extension of their existing business.  
But in 1996 that was a hard sell. The middle managers we talked   
to at catalog companies saw the Web not as an opportunity, but as  
something that meant more work for them.We did get a few of the more adventurous catalog companies. Among  
them was Frederick's of Hollywood, which gave us valuable experience  
dealing with heavy loads on our servers. But most of our users   
were small, individual merchants who saw the Web as an opportunity   
to build a business. Some had retail stores, but many only existed  
online. And so we changed direction to focus on these users.  
Instead of concentrating on the features Web consultants and catalog  
companies would want, we worked to make the software easy to use.I learned something valuable from that. It's worth trying very,   
very hard to make technology easy to use. Hackers are so used to  
computers that they have no idea how horrifying software seems to  
normal people. Stephen Hawking's editor told him that every equation  
he included in his book would cut sales in half. When you work on  
making technology easier to use, you're riding that curve up instead   
of down. A 10% improvement in ease of use doesn't just increase   
your sales 10%. It's more likely to double your sales.How do you figure out what customers want? Watch them. One of the  
best places to do this was at trade shows. Trade shows didn't pay   
as a way of getting new customers, but they were worth it as market  
research. We didn't just give canned presentations at trade shows.  
We used to show people how to build real, working stores. Which   
meant we got to watch as they used our software, and talk to them   
about what they needed.No matter what kind of startup you start, it will probably be a   
stretch for you, the founders, to understand what users want. The  
only kind of software you can build without studying users is the   
sort for which you are the typical user. But this is just the kind  
that tends to be open source: operating systems, programming  
languages, editors, and so on. So if you're developing technology  
for money, you're probably not going to be developing it for people  
like you. Indeed, you can use this as a way to generate ideas for  
startups: what do people who are not like you want from technology?When most people think of startups, they think of companies like  
Apple or Google. Everyone knows these, because they're big consumer  
brands. But for every startup like that, there are twenty more   
that operate in niche markets or live quietly down in the infrastructure.  
So if you start a successful startup, odds are you'll start one of   
those.Another way to say that is, if you try to start the kind of startup  
that has to be a big consumer brand, the odds against succeeding  
are steeper. The best odds are in niche markets. Since startups   
make money by offering people something better than they had before,  
the best opportunities are where things suck most. And it would   
be hard to find a place where things suck more than in corporate   
IT departments. You would not believe the amount of money companies  
spend on software, and the crap they get in return. This imbalance  
equals opportunity.If you want ideas for startups, one of the most valuable things you  
could do is find a middle-sized non-technology company and spend a   
couple weeks just watching what they do with computers. Most good  
hackers have no more idea of the horrors perpetrated in these places  
than rich Americans do of what goes on in Brazilian slums.Start by writing software for smaller companies, because it's easier  
to sell to them. It's worth so much to sell stuff to big companies  
that the people selling them the crap they currently use spend a  
lot of time and money to do it. And while you can outhack Oracle  
with one frontal lobe tied behind your back, you can't outsell an  
Oracle salesman. So if you want to win through better technology,  
aim at smaller customers. [4]They're the more strategically valuable part of the market anyway.   
In technology, the low end always eats the high end. It's easier   
to make an inexpensive product more powerful than to make a powerful  
product cheaper. So the products that start as cheap, simple options  
tend to gradually grow more powerful till, like water rising in a   
room, they squash the "high-end" products against the ceiling. Sun  
did this to mainframes, and Intel is doing it to Sun. Microsoft  
Word did it to desktop publishing software like Interleaf and  
Framemaker. Mass-market digital cameras are doing it to the expensive  
models made for professionals. Avid did it to the manufacturers   
of specialized video editing systems, and now Apple is doing it to  
Avid. Henry Ford did it to the car makers that preceded  
him. If you build the simple, inexpensive option, you'll not only  
find it easier to sell at first, but you'll also be in the best   
position to conquer the rest of the market.It's very dangerous to let anyone fly under you. If you have the  
cheapest, easiest product, you'll own the low end. And if you  
don't, you're in the crosshairs of whoever does.Raising MoneyTo make all this happen, you're going to need money. Some startups  
have been self-funding-- Microsoft for example-- but most aren't.  
I think it's wise to take money from investors. To be self-funding,  
you have to start as a consulting company, and it's hard to switch  
from that to a product company.Financially, a startup is like a pass/fail course. The way to get  
rich from a startup is to maximize the company's chances of succeeding,  
not to maximize the amount of stock you retain. So if you can trade  
stock for something that improves your odds, it's probably a smart   
move.To most hackers, getting investors seems like a terrifying and  
mysterious process. Actually it's merely tedious. I'll try to  
give an outline of how it works.The first thing you'll need is a few tens of thousands of dollars   
to pay your expenses while you develop a prototype. This is called  
seed capital. Because so little money is involved, raising seed  
capital is comparatively easy-- at least in the sense of getting a  
quick yes or no.Usually you get seed money from individual rich people called  
"angels." Often they're people who themselves got rich from technology.  
At the seed stage, investors don't expect you to have an elaborate  
business plan. Most know that they're supposed to decide quickly.  
It's not unusual to get a check within a week based on a half-page  
agreement.We started Viaweb with $10,000 of seed money from our friend Julian.  
But he gave us a lot more than money. He's a former CEO and also  
a corporate lawyer, so he gave us a lot of valuable advice about  
business, and also did all the legal work of getting us set up as  
a company. Plus he introduced us to one of the two   
angel investors who supplied our next round of funding.Some angels, especially those with technology backgrounds, may be   
satisfied with a demo and a verbal description of what you plan to   
do. But many will want a copy of your business plan, if only to  
remind themselves what they invested in.Our angels asked for one, and looking back, I'm amazed how much  
worry it caused me. "Business plan" has that word "business" in  
it, so I figured it had to be something I'd have to read a book  
about business plans to write. Well, it doesn't. At this stage,  
all most investors expect is a brief description of what you plan   
to do and how you're going to make money from it, and the resumes   
of the founders. If you just sit down and write out what you've  
been saying to one another, that should be fine. It shouldn't take  
more than a couple hours, and you'll probably find that writing it  
all down gives you more ideas about what to do.For the angel to have someone to make the check out to, you're going  
to have to have some kind of company. Merely incorporating yourselves  
isn't hard. The problem is, for the company to exist, you have to  
decide who the founders are, and how much stock they each have. If  
there are two founders with the same qualifications who are both  
equally committed to the business, that's easy. But if you have a  
number of people who are expected to contribute in varying degrees,  
arranging the proportions of stock can be hard. And once you've  
done it, it tends to be set in stone.I have no tricks for dealing with this problem. All I can say is,  
try hard to do it right. I do have a rule of thumb for recognizing  
when you have, though. When everyone feels they're getting a  
slightly bad deal, that they're doing more than they should for the  
amount of stock they have, the stock is optimally apportioned.There is more to setting up a company than incorporating it, of  
course: insurance, business license, unemployment compensation,   
various things with the IRS. I'm not even sure what the list is,  
because we, ah, skipped all that. When we got real funding near  
the end of 1996, we hired a great CFO, who fixed everything   
retroactively. It turns out that no one comes and arrests you if  
you don't do everything you're supposed to when starting a company.  
And a good thing too, or a lot of startups would never get started.  
[5]It can be dangerous to delay turning yourself into a company, because  
one or more of the founders might decide to split off and start   
another company doing the same thing. This does happen. So when  
you set up the company, as well as as apportioning the stock, you  
should get all the founders to sign something agreeing that everyone's  
ideas belong to this company, and that this company is going to be  
everyone's only job.[If this were a movie, ominous music would begin here.]While you're at it, you should ask what else they've signed. One  
of the worst things that can happen to a startup is to run into   
intellectual property problems. We did, and it came closer to   
killing us than any competitor ever did.As we were in the middle of getting bought, we discovered that one  
of our people had, early on, been bound by an agreement that said  
all his ideas belonged to the giant company that was paying for him  
to go to grad school. In theory, that could have meant someone  
else owned big chunks of our software. So the acquisition came to  
a screeching halt while we tried to sort this out. The problem   
was, since we'd been about to be acquired, we'd allowed ourselves   
to run low on cash. Now we needed to raise more to keep going.   
But it's hard to raise money with an IP cloud over your head, because  
investors can't judge how serious it is.Our existing investors, knowing that we needed money and had nowhere  
else to get it, at this point attempted certain gambits which I  
will not describe in detail, except to remind readers that the word   
"angel" is a metaphor. The founders thereupon proposed to walk   
away from the company, after giving the investors a brief tutorial   
on how to administer the servers themselves. And while this was  
happening, the acquirers used the delay as an excuse to welch on   
the deal.Miraculously it all turned out ok. The investors backed down; we  
did another round of funding at a reasonable valuation; the giant  
company finally gave us a piece of paper saying they didn't own our  
software; and six months later we were bought by Yahoo for much  
more than the earlier acquirer had agreed to pay. So we were happy  
in the end, though the experience probably took several years off   
my life.Don't do what we did. Before you consummate a startup, ask   
everyone about their previous IP history.Once you've got a company set up, it may seem presumptuous to go  
knocking on the doors of rich people and asking them to invest tens  
of thousands of dollars in something that is really just a bunch   
of guys with some ideas. But when you look at it from the rich  
people's point of view, the picture is more encouraging. Most rich   
people are looking for good investments. If you really think you  
have a chance of succeeding, you're doing them a favor by letting  
them invest. Mixed with any annoyance they might feel about being   
approached will be the thought: are these guys the next Google?Usually angels are financially equivalent to founders. They get   
the same kind of stock and get diluted the same amount in future  
rounds. How much stock should they get? That depends on how  
ambitious you feel. When you offer x percent of your company for  
y dollars, you're implicitly claiming a certain value for the whole  
company. Venture investments are usually described in terms of  
that number. If you give an investor new shares equal to 5% of  
those already outstanding in return for $100,000, then you've done  
the deal at a pre-money valuation of $2 million.How do you decide what the value of the company should be? There  
is no rational way. At this stage the company is just a bet. I  
didn't realize that when we were raising money. Julian  
thought we ought to value the company at several million   
dollars. I thought it was preposterous to claim that a couple  
thousand lines of code, which was all we had at the time, were worth  
several million dollars. Eventually we settled on one million,  
because Julian said no one would invest in a company with a valuation  
any lower. [6]What I didn't grasp at the time was that the valuation wasn't just   
the value of the code we'd written so far. It was also the value  
of our ideas, which turned out to be right, and of all the future  
work we'd do, which turned out to be a lot.The next round of funding is the one in which you might deal with   
actual   
venture capital firms.   
But don't wait till you've burned   
through your last round of funding to start approaching them. VCs are slow to  
make up their minds. They can take months. You don't want to be   
running out of money while you're trying to negotiate with them.Getting money from an actual VC firm is a bigger deal than getting  
money from angels. The amounts of money involved are larger, millions  
usually. So the deals take longer, dilute you more, and impose  
more onerous conditions.Sometimes the VCs want to install a new CEO of their own choosing.   
Usually the claim is that you need someone mature and experienced,  
with a business background. Maybe in some cases this is true. And  
yet Bill Gates was young and inexperienced and had no business   
background, and he seems to have done ok. Steve Jobs got booted  
out of his own company by someone mature and experienced, with a  
business background, who then proceeded to ruin the company. So I  
think people who are mature and experienced, with a business  
background, may be overrated. We used to call these guys "newscasters,"  
because they had neat hair and spoke in deep, confident voices, and  
generally didn't know much more than they read on the teleprompter.We talked to a number of VCs, but eventually we ended up financing  
our startup entirely with angel money. The main reason was that   
we feared a brand-name VC firm would stick us with a newscaster as  
part of the deal. That might have been ok if he was content to  
limit himself to talking to the press, but what if he wanted to   
have a say in running the company? That would have led to disaster,  
because our software was so complex. We were a company whose whole  
m.o. was to win through better technology. The strategic decisions  
were mostly decisions about technology, and we didn't need any help  
with those.This was also one reason we didn't go public. Back in 1998 our CFO  
tried to talk me into it. In those days you could go public as a  
dogfood portal, so as a company with a real product and real revenues,  
we might have done well. But I feared it would have meant taking  
on a newscaster-- someone who, as they say, "can talk Wall Street's  
language."I'm happy to see Google is bucking that trend. They didn't talk  
Wall Street's language when they did their IPO, and Wall Street  
didn't buy. And now Wall Street is collectively kicking itself.  
They'll pay attention next time. Wall Street learns new languages   
fast when money is involved.You have more leverage negotiating with VCs than you realize. The   
reason is other VCs. I know a number of VCs now, and when you talk  
to them you realize that it's a seller's market. Even now there  
is too much money chasing too few good deals.VCs form a pyramid. At the top are famous ones like Sequoia and  
Kleiner Perkins, but beneath those are a huge number you've never   
heard of. What they all have in common is that a dollar from them   
is worth one dollar. Most VCs will tell you that they don't just   
provide money, but connections and advice. If you're talking to   
Vinod Khosla or John Doerr or Mike Moritz, this is true. But such  
advice and connections can come very expensive. And as you go down  
the food chain the VCs get rapidly   
  
dumber. A few steps down from   
the top you're basically talking to bankers who've picked up a few  
new vocabulary words from reading Wired. (Does your product  
use XML?) So I'd advise you to be skeptical about claims  
of experience and connections. Basically, a VC is a source of  
money. I'd be inclined to go with whoever offered the most money   
the soonest with the least strings attached.You may wonder how much to tell VCs. And you should, because some  
of them may one day be funding your competitors. I think the best  
plan is not to be overtly secretive, but not to tell them everything  
either. After all, as most VCs say, they're more interested in the  
people than the ideas. The main reason they want to talk about  
your idea is to judge you, not the idea. So as long as you seem  
like you know what you're doing, you can probably keep a few things  
back from them. [7]Talk to as many VCs as you can, even if you don't want their money,  
because a) they may be on the board of someone who will buy you,   
and b) if you seem impressive, they'll be discouraged from investing  
in your competitors. The most efficient way to reach VCs, especially  
if you only want them to know about you and don't want their money,  
is at the conferences that are occasionally organized for startups   
to present to them.Not Spending ItWhen and if you get an infusion of real money from investors, what  
should you do with it? Not spend it, that's what. In nearly every   
startup that fails, the proximate cause is running out of money.   
Usually there is something deeper wrong. But even a proximate cause  
of death is worth trying hard to avoid.During the Bubble many startups tried to "get big fast." Ideally  
this meant getting a lot of customers fast. But it was easy for  
the meaning to slide over into hiring a lot of people fast.Of the two versions, the one where you get a lot of customers fast  
is of course preferable. But even that may be overrated. The idea  
is to get there first and get all the users, leaving none for  
competitors. But I think in most businesses the advantages of being  
first to market are not so overwhelmingly great. Google is again  
a case in point. When they appeared it seemed as if search was a  
mature market, dominated by big players who'd spent millions to  
build their brands: Yahoo, Lycos, Excite, Infoseek, Altavista,   
Inktomi. Surely 1998 was a little late to arrive at the party.But as the founders of Google knew, brand is worth next to nothing  
in the search business. You can come along at any point and make   
something better, and users will gradually seep over to you. As   
if to emphasize the point, Google never did any advertising. They're  
like dealers; they sell the stuff, but they know better than to use  
it themselves.The competitors Google buried would have done better to spend those  
millions improving their software. Future startups should learn  
from that mistake. Unless you're in a market where products are   
as undifferentiated as cigarettes or vodka or laundry detergent,  
spending a lot on brand advertising is a sign of breakage. And few  
if any Web businesses are so undifferentiated. The dating sites   
are running big ad campaigns right now, which is all the   
more evidence they're ripe for the picking. (Fee, fie, fo, fum, I   
smell a company run by marketing guys.)We were compelled by circumstances to grow slowly, and in retrospect  
it was a good thing. The founders all learned to do every job in   
the company. As well as writing software, I had to do sales and  
customer support. At sales I was not very good. I was persistent,  
but I didn't have the smoothness of a good salesman. My message   
to potential customers was: you'd be stupid not to sell online, and   
if you sell online you'd be stupid to use anyone else's software.   
Both statements were true, but that's not the way to convince people.I was great at customer support though. Imagine talking to a  
customer support person who not only knew everything about the  
product, but would apologize abjectly if there was a bug, and then  
fix it immediately, while you were on the phone with them. Customers  
loved us. And we loved them, because when you're growing slow by  
word of mouth, your first batch of users are the ones who were smart  
enough to find you by themselves. There is nothing more valuable,  
in the early stages of a startup, than smart users. If you listen  
to them, they'll tell you exactly how to make a winning product.   
And not only will they give you this advice for free, they'll pay  
you.We officially launched in early 1996. By the end of that year we  
had about 70 users. Since this was the era of "get big fast," I  
worried about how small and obscure we were. But in fact we were  
doing exactly the right thing. Once you get big (in users or  
employees) it gets hard to change your product. That year was  
effectively a laboratory for improving our software. By the end   
of it, we were so far ahead of our competitors that they never had   
a hope of catching up. And since all the hackers had spent many  
hours talking to users, we understood online commerce way better  
than anyone else.That's the key to success as a startup. There is nothing more   
important than understanding your business. You might think that  
anyone in a business must, ex officio, understand it. Far from it.   
Google's secret  
weapon was simply that they understood search. I was working for   
Yahoo when Google appeared, and Yahoo didn't understand search. I  
know because I once tried to convince the powers that be that we  
had to make search better, and I got in reply what was then the  
party line about it: that Yahoo was no longer a mere "search engine."  
Search was now only a small percentage of our page views, less than  
one month's growth, and now that we were established as a "media   
company," or "portal," or whatever we were, search could safely be  
allowed to wither and drop off, like an umbilical cord.Well, a small fraction of page views they may be, but they are an   
important fraction, because they are the page views that Web sessions   
start with. I think Yahoo gets that now.Google understands a few other things most Web companies still  
don't. The most important is that you should put users before  
advertisers, even though the advertisers are paying and users aren't.  
One of my favorite bumper stickers reads "if the people lead, the   
leaders will follow." Paraphrased for the Web, this becomes "get  
all the users, and the advertisers will follow." More generally,  
design your product to please users first, and then think about how  
to make money from it. If you don't put users first, you leave a   
gap for competitors who do.To make something users love, you have to understand them. And the  
bigger you are, the harder that is. So I say "get big slow." The  
slower you burn through your funding, the more time you have to  
learn.The other reason to spend money slowly is to encourage a culture   
of cheapness. That's something Yahoo did understand. David Filo's   
title was "Chief Yahoo," but he was proud that his unofficial title  
was "Cheap Yahoo." Soon after we arrived at Yahoo, we got an email  
from Filo, who had been crawling around our directory hierarchy,   
asking if it was really necessary to store so much of our data on  
expensive RAID drives. I was impressed by that. Yahoo's market  
cap then was already in the billions, and they were still worrying  
about wasting a few gigs of disk space.When you get a couple million dollars from a VC firm, you tend to  
feel rich. It's important to realize you're not. A rich company  
is one with large revenues. This money isn't revenue. It's money  
investors have given you in the hope you'll be able to generate   
revenues. So despite those millions in the bank, you're still poor.For most startups the model should be grad student, not law firm.  
Aim for cool and cheap, not expensive and impressive. For us the  
test of whether a startup understood this was whether they had Aeron  
chairs. The Aeron came out during the Bubble and was very popular  
with startups. Especially the type, all too common then, that was  
like a bunch of kids playing house with money supplied by VCs. We   
had office chairs so cheap that the arms all fell off. This was   
slightly embarrassing at the time, but in retrospect the grad-studenty  
atmosphere of our office was another of those things we did right   
without knowing it.Our offices were in a wooden triple-decker in Harvard Square. It  
had been an apartment until about the 1970s, and there was still a   
claw-footed bathtub in the bathroom. It must once have been inhabited  
by someone fairly eccentric, because a lot of the chinks in the   
walls were stuffed with aluminum foil, as if to protect against   
cosmic rays. When eminent visitors came to see us, we were a bit  
sheepish about the low production values. But in fact that place  
was the perfect space for a startup. We felt like our role was to   
be impudent underdogs instead of corporate stuffed shirts, and that   
is exactly the spirit you want.An apartment is also the right kind of place for developing software.  
Cube farms suck for that, as you've probably discovered if you've  
tried it. Ever notice how much easier it is to hack at home than  
at work? So why not make work more like home?When you're looking for space for a startup, don't feel that it has  
to look professional. Professional means doing good work, not  
elevators and glass walls. I'd advise most startups to avoid  
corporate space at first and just rent an apartment. You want to  
live at the office in a startup, so why not have a place designed  
to be lived in as your office?Besides being cheaper and better to work in, apartments tend to be  
in better locations than office buildings. And for a startup  
location is very important. The key to productivity is for people  
to come back to work after dinner. Those hours after the phone  
stops ringing are by far the best for getting work done. Great  
things happen when a group of employees go out to dinner together,   
talk over ideas, and then come back to their offices to implement   
them. So you want to be in a place where there are a lot of  
restaurants around, not some dreary office park that's a wasteland  
after 6:00 PM. Once a company shifts over into the model where   
everyone drives home to the suburbs for dinner, however late, you've  
lost something extraordinarily valuable. God help you if you  
actually start in that mode.If I were going to start a startup today, there are only three   
places I'd consider doing it: on the Red Line near Central, Harvard,  
or Davis Squares (Kendall is too sterile); in Palo Alto on University  
or California Aves; and in Berkeley immediately north or south of   
campus. These are the only places I know that have the right kind  
of vibe.The most important way to not spend money is by not hiring people.   
I may be an extremist, but I think hiring people is the worst thing  
a company can do. To start with, people are a recurring expense,   
which is the worst kind. They also tend to cause you to grow out   
of your space, and perhaps even move to the sort of uncool office  
building that will make your software worse. But worst of all,  
they slow you down: instead of sticking your head in someone's   
office and checking out an idea with them, eight people have to  
have a meeting about it. So the fewer people you can hire, the  
better.During the Bubble a lot of startups had the opposite policy. They  
wanted to get "staffed up" as soon as possible, as if you couldn't   
get anything done unless there was someone with the corresponding   
job title. That's big company thinking. Don't hire people to fill  
the gaps in some a priori org chart. The only reason to hire someone  
is to do something you'd like to do but can't.If hiring unnecessary people is expensive and slows you down, why  
do nearly all companies do it? I think the main reason is that  
people like the idea of having a lot of people working for them.  
This weakness often extends right up to the CEO. If you ever end  
up running a company, you'll find the most common question people  
ask is how many employees you have. This is their way of weighing  
you. It's not just random people who ask this; even reporters do.  
And they're going to be a lot more impressed if the answer is a  
thousand than if it's ten.This is ridiculous, really. If two companies have the same revenues,  
it's the one with fewer employees that's more impressive. When   
people used to ask me how many people our startup had, and I answered  
"twenty," I could see them thinking that we didn't count for much.  
I used to want to add "but our main competitor, whose ass we regularly  
kick, has a hundred and forty, so can we have credit for the larger  
of the two numbers?"As with office space, the number of your employees is a choice   
between seeming impressive, and being impressive. Any of you who   
were nerds in high school know about this   
choice. Keep doing it when you start a company.Should You?But should you start a company? Are you the right sort of person  
to do it? If you are, is it worth it?More people are the right sort of person to start a startup than  
realize it. That's the main reason I wrote this. There could be   
ten times more startups than there are, and that would probably be  
a good thing.I was, I now realize, exactly the right sort of person to start a   
startup. But the idea terrified me at first. I was forced into   
it because I was a Lisp hacker. The company  
I'd been consulting for seemed to be running into trouble, and there   
were not a lot of other companies using Lisp. Since I couldn't   
bear the thought of programming in another language (this was 1995,  
remember, when "another language" meant C++) the only option seemed  
to be to start a new company using Lisp.I realize this sounds far-fetched, but if you're a Lisp hacker  
you'll know what I mean. And if the idea of starting a startup  
frightened me so much that I only did it out of necessity, there   
must be a lot of people who would be good at it but who are too   
intimidated to try.So who should start a startup? Someone who is a good hacker, between  
about 23 and 38, and who wants to solve the money problem in one  
shot instead of getting paid gradually over a conventional working  
life.I can't say precisely what a good hacker is. At a first rate   
university this might include the top half of computer science   
majors. Though of course you don't have to be a CS major to be a  
hacker; I was a philosophy major in college.It's hard to tell whether you're a good hacker, especially when  
you're young. Fortunately the process of starting startups tends  
to select them automatically. What drives people to start startups  
is (or should be) looking at existing technology and thinking, don't   
these guys realize they should be doing x, y, and z? And that's   
also a sign that one is a good hacker.I put the lower bound at 23 not because there's something that  
doesn't happen to your brain till then, but because you need to see  
what it's like in an existing business before you try running your  
own. The business doesn't have to be a startup. I spent a year  
working for a software company to pay off my college loans. It was  
the worst year of my adult life, but I learned, without realizing   
it at the time, a lot of valuable lessons about the software business.  
In this case they were mostly negative lessons: don't have a lot  
of meetings; don't have chunks of code that multiple people own;  
don't have a sales guy running the company; don't make a high-end  
product; don't let your code get too big; don't leave finding bugs  
to QA people; don't go too long between releases; don't isolate  
developers from users; don't move from Cambridge to Route 128; and  
so on. [8] But negative lessons are just as valuable as positive   
ones. Perhaps even more valuable: it's hard to repeat a brilliant  
performance, but it's straightforward to avoid errors. [9]The other reason it's hard to start a company before 23 is that   
people won't take you seriously. VCs won't trust you, and will try  
to reduce you to a mascot as a condition of funding. Customers  
will worry you're going to flake out and leave them stranded. Even  
you yourself, unless you're very unusual, will feel your age to   
some degree; you'll find it awkward to be the boss of someone much   
older than you, and if you're 21, hiring only people younger rather   
limits your options.Some people could probably start a company at 18 if they wanted to.  
Bill Gates was 19 when he and Paul Allen started Microsoft. (Paul   
Allen was 22, though, and that probably made a difference.) So if  
you're thinking, I don't care what he says, I'm going to start a  
company now, you may be the sort of person who could get away with  
it.The other cutoff, 38, has a lot more play in it. One reason I put   
it there is that I don't think many people have the physical stamina  
much past that age. I used to work till 2:00 or 3:00 AM every  
night, seven days a week. I don't know if I could do that now.Also,  
startups are a big risk financially. If you try something that  
blows up and leaves you broke at 26, big deal; a lot of 26 year  
olds are broke. By 38 you can't take so many risks-- especially  
if you have kids.My final test may be the most restrictive. Do you actually want  
to start a startup? What it amounts to, economically, is compressing  
your working life into the smallest possible space. Instead of  
working at an ordinary rate for 40 years, you work like hell for   
four. And maybe end up with nothing-- though in that case it  
probably won't take four years.During this time you'll do little but work, because when you're not  
working, your competitors will be. My only leisure activities were  
running, which I needed to do to keep working anyway, and about  
fifteen minutes of reading a night. I had a girlfriend for a total  
of two months during that three year period. Every couple weeks I  
would take a few hours off to visit a used bookshop or go to a   
friend's house for dinner. I went to visit my family twice.  
Otherwise I just worked.Working was often fun, because the people I worked with were some  
of my best friends. Sometimes it was even technically interesting.  
But only about 10% of the time. The best I can say for the other  
90% is that some of it is funnier in hindsight than it seemed then.  
Like the time the power went off in Cambridge for about six hours,  
and we made the mistake of trying to start a gasoline powered  
generator inside our offices. I won't try that again.I don't think the amount of bullshit you have to deal with in a  
startup is more than you'd endure in an ordinary working life. It's  
probably less, in fact; it just seems like a lot because it's   
compressed into a short period. So mainly what a startup buys you   
is time. That's the way to think about it if you're trying to   
decide whether to start one. If you're the sort of person who would  
like to solve the money problem once and for all instead of working   
for a salary for 40 years, then a startup makes sense.For a lot of people the conflict is between startups and graduate  
school. Grad students are just the age, and just the sort of people,  
to start software startups. You may worry that if you do you'll   
blow your chances of an academic career. But it's possible to be   
part of a startup and stay in grad school, especially at first.   
Two of our three original hackers were in grad school the whole   
time, and both got their degrees.   
There are few sources of energy  
so powerful as a procrastinating grad student.If you do have to  
leave grad school, in the worst case it won't be for too long. If  
a startup fails, it will probably fail quickly enough that you can   
return to academic life. And if it succeeds, you may find you no   
longer have such a burning desire to be an assistant professor.If you want to do it, do it. Starting a startup is not the great  
mystery it seems from outside. It's not something you have to know  
about "business" to do. Build something users love, and spend less  
than you make. How hard is that?Notes[1] Google's revenues are about two billion a year, but half comes  
from ads on other sites.[2] One advantage startups have over established companies is that  
there are no discrimination laws about starting businesses. For   
example, I would be reluctant to start a startup with a woman  
who had small children, or was likely to have them soon. But you're  
not allowed to ask prospective employees if they plan to have kids   
soon. Believe it or not, under current US law, you're not even   
allowed to discriminate on the basis of intelligence. Whereas when  
you're starting a company, you can discriminate on any basis you  
want about who you start it with.[3] Learning to hack is a lot cheaper than business school, because  
you can do it mostly on your own. For the price of a Linux box, a  
copy of K&R, and a few hours of advice from your neighbor's fifteen  
year old son, you'll be well on your way.[4] Corollary: Avoid starting a startup to sell things to the biggest  
company of all, the government. Yes, there are lots of opportunities  
to sell them technology. But let someone else start those startups.[5] A friend who started a company in Germany told me they do care   
about the paperwork there, and that there's more of it. Which helps  
explain why there are not more startups in Germany.[6] At the seed stage our valuation was in principle $100,000, because  
Julian got 10% of the company. But this is a very misleading number,  
because the money was the least important of the things Julian gave us.[7] The same goes for companies that seem to want to acquire you.  
There will be a few that are only pretending to in order to pick  
your brains. But you can never tell for sure which these are, so  
the best approach is to seem entirely open, but to fail to mention  
a few critical technical secrets.[8] I was as bad an employee as this place was a company. I  
apologize to anyone who had to work with me there.[9] You could probably write a book about how to succeed in business  
by doing everything in exactly the opposite way from the DMV.Thanks to Trevor Blackwell, Sarah Harlin, Jessica Livingston,  
and Robert Morris for reading drafts of this essay, and to Steve  
Melendez and Gregory Price for inviting me to speak.