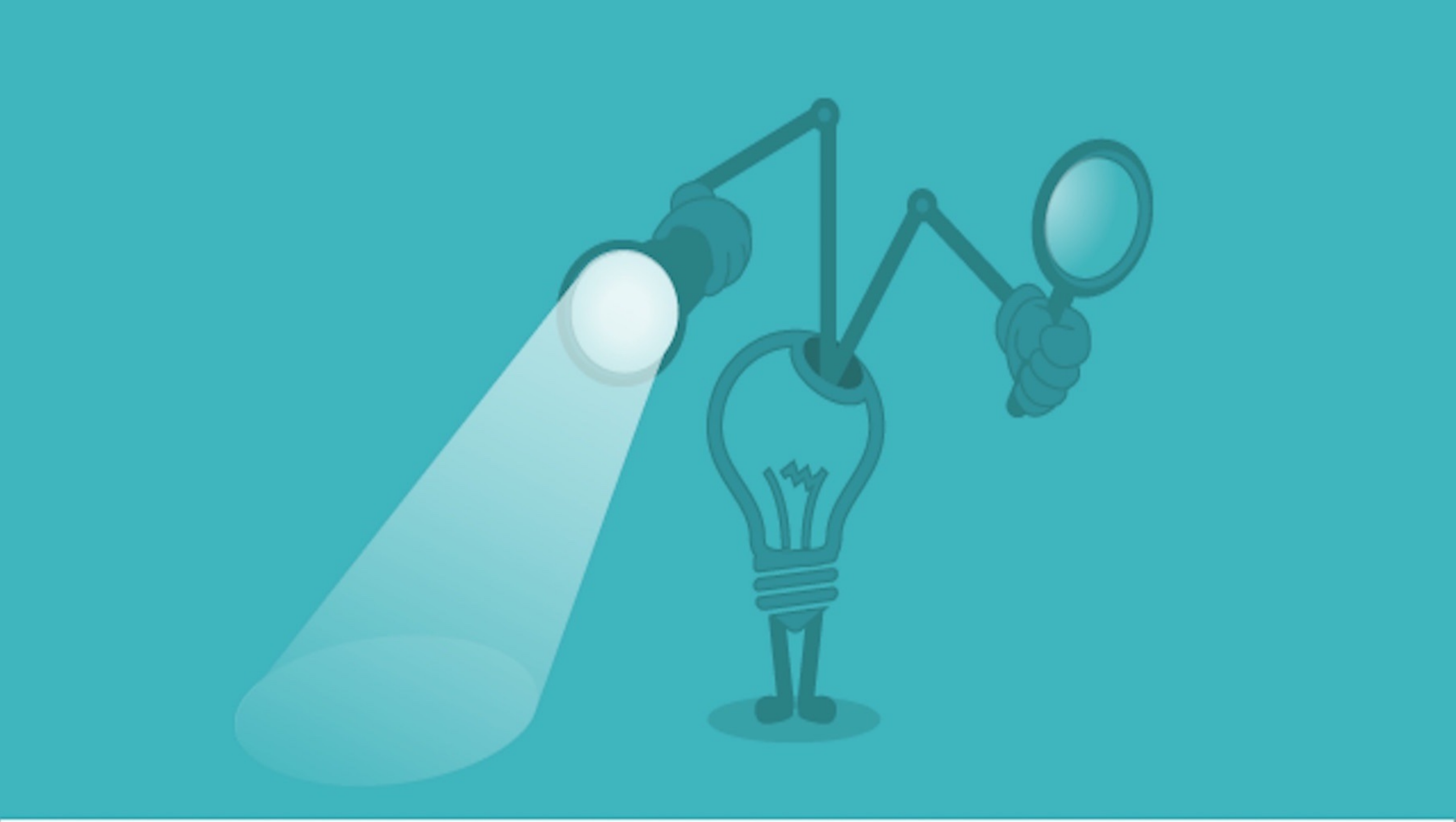

How to Start a Startup

A YC & Stanford Course

Table of Contents

How to Start a Startup

-- A YC and Stanford Course

Everything we know about how to start a startup, for free, from some of the world experts.

> - The Course is **FREE**, and available at: https://startupclass.co.
> - Lecture Videos also available at youtube.
> - iTunes Podcast available here.

Description

Sam Altman and the folks from Y Combinator offer up an amazing course in "How To Start A Startup" at Stanford. Course includes lectures from: Sam Altman, Dustin Moskovitz, Paul Graham, Adora Cheung, Peter Thiel, Alex Schultz, Kevin Hale, Marc Andreessen, Ron Conway, Ben Silbermann, Alfred Lin, Patrick and John Collison, Aaron Levie, Reid Hoffman, Keith Rabois, Ben Horowitz, Marissa Mayer, Hosian Rahman, Kirsty Nathoo, Carolynn Levy, and more.

The trasncript was taken from geius.com. I created this book for those who prefer reading over watching for the sake of efficiency.

xiaolai's Notes

I wrote numerous notes in Chinese here: http://zhibimo.com/explore/growth.

A letter from Sam Altman, President at YCombinator about the course:

> CS183B is a class we're teaching at Stanford. It's designed to be a sort of one-class business course for people who want to start startups.
>
> Videos of the lectures, associated reading materials, and assignments will all be available here. There will be 20 videos, some with a speaker or two and some with a small panel. It'll be 1,000 minutes of content if you watch it all.
>
> We'll cover how to come up with ideas and evaluate them, how to get users and grow, how to do sales and marketing, how to hire, how to raise money, company culture, operations and management, business strategy, and more.
>
> You can't teach everything necessary to succeed in starting a company, but I suspect we can teach a surprising amount. We've tried to take some of the best speakers from the past 9 years of Y Combinator dinners and arrange them in a way that will hopefully make sense.
>
> We're doing this because we believe helping a lot of people be better at starting companies will be good for everyone. It will hopefully be valuable even for people who don't want to start startups. Talks like these have really helped Y Combinator founders create their companies. We hope you find it helpful too!
>
> ***-Sam***

Table of Content

Lecture 1: How to Start a Startup

Sam Altman

Welcome to CS183B. I am Sam Altman, I'm the President of Y Combinator. Nine years ago, I was a Stanford student, and then I dropped out to start a company and then I've been an investor for the last few. So YC, we've been teaching people how to start startups for nine years. Most of it's pretty specific to the startups but thirty percent of it is pretty generally applicable. And so we think we can teach that thirty percent in this class. And even though that's only thirty percent of the way there, hopefully it will still be really helpful.

We've taught a lot of this class at YC and it's all been off the record. And this is the first time a lot of what we teach is going to be on the record. We've invited some of our guest speakers to come and give the same talks they give at YC. We've now funded 725 companies and so we're pretty sure a lot of this advice we give is pretty good. We can't fund every startup yet, but we can hopefully make this advice very generally available.

I'm only teaching three. Counting YC itself, every guest speaker has been involved in the creation of a billion plus dollar company. So the advice shouldn't be that theoretical, it's all been people who have done it.

All of the advice in this class is geared towards people starting a business where the goal is hyper growth and eventually building a very large company. Much of it doesn't apply in other cases and I want to warn people up front, that if you try to do these things in a lot of big companies or non-startups, it won't work. It should still be interesting, I really think that startups are the way of the future and it's worth trying to understand them, but startups are very different than normal companies. So over the course of today and Thursday, I'm going to try to give an overview of the four areas you need to excel at in order to maximize your success as a startup. And then throughout the course, the guest speakers are going to drill into all of these in more detail.

Ideas, Products, Teams and Execution Part I

So the four areas: You need a great idea, a great product, a great team, and great execution. These overlap somewhat, but I'm going to have to talk about them somewhat individually to make it make sense.

You may still fail. The outcome is something like idea x product x execution x team x luck, where luck is a random number between zero and ten thousand. Literally that much. But if you do really well in the four areas you can control, you have a good chance at at least some amount of success.

One of the exciting things about startups is that they are a surprisingly even playing field. Young and inexperienced, you can do this. Old and experienced, you can do this, too. And one of the things that I particularly like about startups is that some of the things that are bad in other work situations, like being poor and unknown, are actually huge assets when it comes to starting a startup.

Before we jump in on the how, I want to talk about why you should start a startup. I'm somewhat hesitant to be doing this class at all because you should never start a startup just for the sake of doing so. There are much easier ways to become rich and everyone who starts a startup always says, always, that they couldn't have imagined how hard and painful it was going to be. You should only start a startup if you feel compelled by a particular problem and that you think starting a company is the best way to solve it.

The specific passion should come first, and the startup second. In fact, all of the classes we have at YC follow this. So for the second half of today's lecture, Dustin Moskovitz is going to take over and talk about why to start a startup. We were so surprised at the amount of attention this class got, that we wanted to make sure we spent a lot of time on the why.

The first of the four areas: a great idea. It's become popular in recent years to say that the idea doesn't matter. In fact, it's uncool to spend a lot of time thinking about the idea for a startup. You're just supposed to start, throw stuff at the wall, see what sticks, and not even spend any time thinking about if it will be valuable if it works.

And pivots are supposed to be great, the more pivots the better. So this isn't totally wrong, things do evolve in ways you can't totally predict. And there's a limit to how much you can figure out without actually getting a product in the hands of the users. And great execution is at least ten times as important and a hundred times harder than a great idea.

But the pendulum has swung way out of whack. A bad idea is still bad and the pivot-happy world we're in today feels suboptimal. Great execution towards a terrible idea will get you nowhere. There are exceptions, of course, but most great companies start with a great idea, not a pivot.

If you look at successful pivots, they almost always are a pivot into something the founders themselves wanted, not a random made up idea. Airbnb happened because Brian Chesky couldn't pay his rent, but he had some extra space. In general though if you look at the track record of pivots, they don't become big companies. I myself used to believe ideas didn't matter that much, but I'm very sure that's wrong now.

The definition of the idea, as we talk about it, is very broad. It includes the size and the growth of the market, the growth strategy for the company, the defensibility strategy, and so on. When you're evaluating an idea, you need to think through all these things, not just the product. If it works out, you're going to be working on this for ten years so it's worth some real up front time to think through the up front value and the defensibility of the business. Even though plans themselves are worthless, the exercise of planning is really valuable and totally missing in most startups today.

Long-term thinking is so rare anywhere, but especially in startups. There is a huge advantage if you do it. Remember that the idea will expand and become more ambitious as you go. You certainly don't need to have everything figured out in your path to world domination, but you really want a nice kernel to start with. You want something that can develop in interesting ways.

As you're thinking through ideas, another thing we see that founders get wrong all the time is that someday you need to build a business that is difficult to replicate. This is an important part of a good idea.

I want to make this point again because it is so important: the idea should come first and the startup should come second. Wait to start a startup until you come up with an idea you feel compelled to explore. This is also the way to choose between ideas. If you have several ideas, work on the one that you think about most often when you're not trying to think about work. What we hear again and again from founders is that they wish they had waited until they came up with an idea they really loved.

Another way of looking at this is that the best companies are almost always mission oriented. It's difficult to get the amount of focus that large companies need unless the company feels like it has an important mission. And it's usually really hard to get that without a great founding idea. A related advantage of mission oriented ideas is that you yourself will be dedicated to them. It takes years and years, usually a decade, to build a great startup. If you don't love and believe in what you're building, you're likely to give up at some point along the way. There's no way I know of to get through the pain of a startup without the belief that the mission really matters. A lot of founders, especially students, believe that their startups will only take two to three years and then after that they'll work on what they're really passionate about. That almost never works. Good startups usually take ten years.

A third advantage of mission oriented companies is that people outside the company are more willing to help you. You'll get more support on a hard, important project, than a derivative one. When it comes to starting a startup, it's easier to found a hard startup than an easy startup. This is one of those counter-intuitive things that takes people a long time to understand. It's difficult to overstate how important being mission driven is, so I want to state it one last time: derivative companies, companies that copy an existing idea with very few new insights, don't excite people and they don't compel the teams to work hard enough to be successful.

Paul Graham is going to talk about how to get startup ideas next week. It's something that a lot of founders struggle with, but it's something I believe you can get better at with practice and it's definitely worth trying to get better at.

The hardest part about coming up with great ideas, is that the best ideas often look terrible at the beginning. The thirteenth search engine, and without all the features of a web portal? Most people thought that was pointless. Search was done, and anyways, it didn't matter that much. Portals were where the value was at. The tenth social network, and limited only to college students with no money? Also terrible. MySpace has won and who wants college students as customers? Or a way to stay on strangers' couches. That just sounds terrible all around.

These all sounded really bad but they turned out to be good. If they sounded really good, there would be too many people working on them. As Peter Thiel is going to discuss in the fifth class, you want an idea that turns into a monopoly. But you can't get a monopoly right away. You have to find a small market in which you can get a monopoly and then quickly expand. This is why some great startup ideas look really bad at the beginning. It's good if you can say something like, "Today, only this small subset of users are going to use my product, but I'm going to get all of them, and in the future, almost everyone is going to use my product."

Here is the theme that is going to come up a lot: you need conviction in your own beliefs and a willingness to ignore others' naysaying. The hard part is that this is a very fine line. There's right on one side of it, and crazy on the other. But keep in mind that if you do come up with a great idea, most people are going to think it's bad. You should be happy about that, it means they won't compete with you.

That's why it's also not dangerous to tell people your idea. The truly good ideas don't sound like they're worth stealing. You want an idea where you can say, "I know it sounds like a bad idea, but here's specifically why it's actually a great one." You want to sound crazy, but you want to actually be right. And you want an idea that not many other people are working on. And it's okay if it doesn't sound big at first.

A common mistake among founders, especially first time founders, is that they think the first version of their product - the first version of their idea - needs to sound really big. But it doesn't. It needs to take over a small specific market and expand from there. That's how most great companies get started. Unpopular but right is what you're going for. You want something that sounds like a bad idea, but is a good idea.

You also really want to take the time to think about how the market is going to evolve. You need a market that's going to be big in 10 years. Most investors are obsessed with the market size today, and they don't think at all about how the market is going to evolve.

In fact, I think this is one of the biggest systemic mistakes that investors make. They think about the growth of the start-up itself, they don't think about the growth of the market. I care much more about the growth rate of the market than its current size, and I also care if there's any reason it's going to top out. You should think about this. I prefer to invest in a company that's going after a small, but rapidly growing market, than a big, but slow-growing market.

One of the big advantages of these sorts of markets - these smaller, rapidly growing markets - is that customers are usually pretty desperate for a solution, and they'll put up with an imperfect, but rapidly improving product. A big advantage of being a student - one of the two biggest advantages - is that you probably have better intuition about which markets are likely to start growing rapidly than older people do. Another thing that students usually don't understand, or it takes awhile, [is that] you can not create a market that does not exist. You can basically change everything in a start-up but the market, so you should actually do some thinking to be sure - or be as sure as you can be - that the market you're going after is going to grow and be there.

There are a lot of different ways to talk about the right kind of market. For example, surfing some one else's wave, stepping into an up elevator, or being part of a movement, but all of this is just a way of saying that you want a market that's going to grow really quickly. It may seem small today, it may be small today, but you know - and other people don't - that it's going to grow really fast.

So think about where this is happening in the world. You need this sort of tailwind to make a startup successful.

The exciting thing is the there are probably more of these tailwinds now then ever before. As Marc Andreessen says, software is eating the world. Its just everywhere, there are so many great ideas out there. You just have to pick one, and find one that you really care about.

Another version of this, that gets down to the same idea, is Sequoia's famous question: Why now? Why is this the perfect time for this particular idea, and to start this particular company. Why couldn't it be done two years ago, and why will two years in the future be two late? For the most successful startups we've been involved with, they've all had a great idea and a great answer to this question. And if you don't you should be at least somewhat suspicious about it.

In general, its best if you're building something that you yourself need. You'll understand it much better than if you have to understand it by talking to a customer to build the very first version. If you don't need it yourself, and you're building something someone else needs, realize that you're at a big disadvantage, and get very very close to your customers. Try to work in their office, if you can, and if not, talk to them multiple times a day.

Another somewhat counterintuitive thing about good startup ideas is that they're almost always very easy to explain and very easy to understand. If it takes more then a sentence to explain what you're doing, that's almost always a sign that its too complicated. It should be a clearly articulated vision with a small number of words. And the best ideas are usually very different from existing companies, [either] in one important way, like Google being a search engine that worked just really well, and none of the other stuff of the portals, or totally new, like SpaceX. Any company that's a clone of something else, that already exists, with some small or made up differentiator—like X, beautiful design, or Y for people that like red wine instead—that usually fails.

So as I mentioned, one of the great things about being a student is that you've got a very good perspective on new technology. And learning to have good ideas takes a while, so start working on that right now. That's one thing we hear from people all the time, that they wish they had done more of as a student.

The other is meeting potential cofounders. You have no idea how good of an environment you're in right now, for meeting people you can start a company with down the road. And the one thing that we always tell college students is that more important then any particular startup is getting to know potential cofounders.

So I want to finish this section of my talk with a quote from 50 Cent. This is from when he was asked about Vitamin Water. I won't read it, it's up there, but it's about the importance of thinking about what customers want, and thinking about the demands of the market. Most people don't do this—most students especially don't do this. If you can just do this one thing, if you can just learn to think about the market first, you'll have a big leg up on most people starting startups. And this is probably the thing we see wrong with Y Combinator apps most frequently, is that people have not thought about the market first, and what people want first.

So for the next section, I'm going to talk about building a great product. And here, again, I'm going to use a very broad definition of product. It includes customer support, the copy you write explaining the product, anything involved in your customer's interaction in what you built for them.

To build a really great company, you first have to turn a great idea into a great product. This is really hard, but its crucially important, and fortunately its pretty fun. Although great products are always new to the world, and its hard to give you advice about what to build, there are enough commonalities that we can give you a lot of advice about how to build it.

One of the most important tasks for a founder is to make sure that the company builds a great product. Until you build a great product, nothing else matters. When really successful startup founders tell the story of their early days its almost always sitting in front of the computer working on their product, or talking to their customers. That's pretty much all the time. They do very little else, and you should be very skeptical if your time allocation is much different. Most other problems that founders are trying to solve, raising money, getting more press, hiring, business development, et cetera, these are significantly easier when you have a great product. Its really important to take care of that first. Step one is to build something that users love. At YC, we tell founders to work on their product, talk to users, exercise, eat and sleep, and very little else. All the other stuff I just mentioned—PR, conferences, recruiting advisers, doing partnerships—you should ignore all of that, and just build a product and get it as good as possible by talking to your users.

Your job is to build something that users love. Very few companies that go on to be super successful get there without first doing this. A lot of good-on-paper startups fail because they merely make something that people like. Making something that people want, but only a medium amount, is a great way to fail, and not understand why you're failing. So these are the two jobs

Something that we say at YC a lot is that its better to build something that a small number of users love, then a large number of users like. Of course, it would be best to build something that a small number of users love, but opportunities to do that for v1 are rare, and they're usually not available to startups. So in practice you end up choosing the gray of the orange. You make something that a lot of users like a little bit, or something that a small number of users love a lot. This is a very important piece of advice. Build something that a small number of users love. It is much easier to expand from something that small number of people love, to something that a lot of people love, then from something that a lot of people like to a lot of people love. If you get right, you can get a lot of other things wrong. If you don't get this right, you can get everything else right, and you'll probably still fail. So when you start on the startup, this is the only thing you need to care about until its working.

[Audience member]: Can you go over that slide again?

So you have a choice in a startup. The best thing of all worlds is to build a product that a lot of people really love. In practice, you can't usually do that, because if there's an opportunity like that, Google or Facebook will do it. So there's like a limit to the area under the curve, of what you can build. So you can build something that a large number of users like a little bit, or a small number of users love a lot. So like the total amount of love is the same, its just a question of how its distributed. [audience laughter] And there's like this law of conservation of how much happiness you can put in the world, with the first product of a startup.

And so startups always struggle, with which of those two they should go. And they seem equal, right? Because the area under the curve is the same. But we've seen this time and again, that they're not. And that it's so much easier to expand, once you've got something that some people love, you can expand that into something that a lot of other people love. But if you start with ambivalence, or weak enthusiasm, and try to expand that, you'll never get up to a lot of people loving it. So the advice is: find a small group of users, and make them love what you're doing

One way that you know when this is working, is that you'll get growth by word of mouth. If you get something people love, people will tell their friends about it. This works for consumer product and enterprise products as well. When people really love something, they'll tell their friends about it, and you'll see organic growth.

If you find yourself talking about how it's okay that you're not growing—because there's a big partnership that's going to come save you or something like that—its almost always a sign of real trouble. Sales and marketing are really important, and we're going to have two classes on them later. A great product is the secret to long term growth hacking. You should get that right before anything else. It doesn't get easier to put off making a great product. If you try to build a growth machine before you have a product that some people really love, you're almost certainly going to waste your time. Breakout companies almost always have a product that's so good, it grows by word of mouth. Over the long run, great product win. Don't worry about your competitors raising a lot of money, or what they might do in the future. They probably aren't very good anyway. Very few startups die from competition. Most die because they themselves fail to make something users love, they spend their time on other things. So worry about this about all else.

Another piece of advice to make something that users love: start with something simple. Its much much easier to make a great product if you have something simple. Even if your eventual plans are super complex, and hopefully they are, you can almost always start with a smaller subset of the problem then you think is the smallest, and its hard to build a great product, so you want to start with as little surface area as possible. Think about the really successful companies, and what they started with, think about products you really love. They're generally incredibly simple to use, and especially to get started using. The first version of Facebook was almost comically simple. The first version of Google was just a webpage with a textbox and two buttons; but it returned the best results, and that's why users loved it. The iPhone is far simpler to use then any smartphone that ever came before it, and it was the first one users really loved.

Another reason that simple's good is because it forces you to do one thing extremely well and you have to do that to make something that people love.

The word fanatical comes up again and again when you listen to successful founders talk about how they think about their product. Founders talk about being fanatical in how they care about the quality of the small details. Fanatical in getting the copy that they use to explain the product just right. and fanatical in the way that they think about customer support. In fact, one thing that correlates with success among the YC companies is the founders that hook up Pagerduty to their ticketing system, so that even if the user emails in the middle of the night when the founder's asleep, they still get a response within an hour. Companies actually do this in the early days. Their founders feel physical pain when the product sucks and they want to wake up and fix it. They don't ship crap, and if they do, they fix it very very quickly. And it definitely takes some level of fanaticism to build great products.

You need some users to help with the feedback cycle, but the way you should get those users is manually—you should go recruit them by hand. Don't do things like buy Google ads in the early days, to get initial users. You don't need very many, you just need ones that will give you feedback everyday, and eventually love your product. So instead of trying to get them on Google Adwords, just the few people, in the world, that would be good users. Recruit them by hand.

Ben Silbermann, when everyone thought Pinterest was a joke, recruited the initial Pinterest users by chatting up strangers in coffee shops. He really did, he just walked around Palo Alto and said "Will you please use my product?" He also used to run around the Apple store in Palo Alto, and he would like set all the browsers to the Pinterest homepage real quick, before they caught him and kicked him out, (laughter) and so that when people walked in they were like "Oh, what's this?". This is an important example of doing things that don't scale. If you haven't read Paul Graham's essay on that topic, you definitely should.

So get users manually and remember that the goal is to get a small group of them to love you. Understand that group extremely well, get extremely close to them. Listen to them and you'll almost always find out that they're very willing to give you feedback. Even if you're building the product for yourself, listen to outside users, and they'll tell you how to make a product they'll pay for. Do whatever you need to make them love you, and make them know what you're doing. Because they'll also be the advocates that help you get your next users.

You want to build an engine in the company that transforms feedback from users into product decisions. Then get it back in from of the users and repeat. Ask them what the like and don't like, and watch them use it. Ask them what they'd pay for. Ask them if they'd be really bummed if your company went away. Ask them what would make them recommend the product to their friends, and ask them if they'd recommended it to any yet.

You should make this feedback loop as tight as possible. If your product gets 10 percent better every week, that compounds really quickly. One of the advantages of software startups is just how short you can make the feedback loop. It can be measured in hours, and the best companies usually have the tightest feedback loop. You should try to keep this going for all of your company's life, but its really important in the early days.

The good news is that all this is doable. Its hard, it takes a lot of effort, but there's no magic. The plan is at least is straightforward, and you will eventually get to a great product.

Great founders don't put anyone between themselves and their users. The founders of these companies do things like sales and customer support themselves in the early days. Its critical to get this loop embedded in the culture. In fact, a specific problem we always see with Stanford startups, for some reason, is that the students try to hire sales and customer support people right away, and you've got to do this yourself, its the only way.

You really need to use metrics to keep yourself honest on this. It really is true that the company will build whatever the CEO decides to measure. If you're building an Internet service, ignore things like total registrations—don't talk about them, don't let anyone in the company talk about them—and look at growth and active users, activity levels, cohort retention, revenue, net promoter scores, these things that matter. And then be brutally honest if they're not going in the right direction. Startups live on growth, its the indicator of a great product.

So this about wraps up the overview on building a great product. I want to emphasize again, that if you don't get this right, nothing else we talk about in the class will matter. You can basically ignore everything else in the class until this is working well. On the positive side, this is one of the most fun parts of building a startup.

So I'm going to pause here, we'll pick back up with the rest of this on Thursday, and now Dustin is going to talk about why you should start a startup. Thank you for coming, Dustin.

Why To Start A Startup

But yeah, Sam asked me to talk about why you should start a startup. There's a bunch of common reasons that people have, that I hear all the time for why you might start a startup. Its important to know what reason is yours, because some of them only make sense in certain contexts, some of them will actually, like, lead you astray. You may have been mislead by the way that Hollywood or the press likes to romanticize entrepreneurship, so I want to try to illuminate some of those potential fallacies, so you guys can make the decision in a clear way. And then I'll talk about the reason I like best for actually starting a startup, its very related to a lot of what Sam just talked about. But surprisingly, I don't think its the most common reason. Usually people have one of these other reasons, or, you know, they just want to start a company for the sake of starting a company.

So the 4 common reasons, just to enumerate them, are it's glamorous, you'll get to be the boss, you'll have flexibility, especially over your schedule, and you'll have the chance to have bigger impact and make more money then you might by joining a later stage company.

So you guys are probably pretty familiar this concept, when I wrote the Medium post, which a lot of you guys read a year ago, I felt like the story in the press was a little more unbalanced, entrepreneurship got romanticized quite a bit. The movie The Social Network came out, it had a lot of like bad aspects of what it like to be an entrepreneur, but mainly it painted this picture of like, there's a lot of partying and you just kind of move from like one brilliant insight to another brilliant insight, and really made it seem like this really cool thing to do.

And I think the reality is just not quite so glamorous, there's an ugly side to being an entrepreneur, and more importantly, what you're actually spending your time on is just a lot of hard work. Sam mentioned this, but your basically just sitting at your desk, heads down, focused, answering customer support emails, doing sales, figuring out hard engineering problems. So its really important that you go in with eyes wide open. And then its also quite stressful. This has been a popular topic in the press lately: The Economist actually ran a story just last week called "Entreupeneurs anonymous", and shows a founder like hiding under his desk, talking about founder depression. So this is a very real thing. Let's be real, if you start a company its going to be extremely hard.

Why is it so stressful? So a couple reasons. One is you've got a lot of responsibility. People in any career have a fear of failure, its kind of just like a dominant part of the part of the psychology. But when you're an entrepreneur, you have fear of failure on behalf of yourself and all of the people who decided to follow you. So that's really stressful. In some cases people are depending on you for their livelihood, even when that's not true, they've decided to devote the best years of their life to following you. So you're responsible for the opportunity cost of their time. You're always on call, if something comes up—maybe not always at 3 in the morning, but for some startups that's true—but if something important comes up, you're going to deal with it. That's kinda the end of the story, doesn't matter if you're on vacation, doesn't matter if its the weekend, you've got to always be on the ball and be in a place mentally where you're prepared to deal with those things. A sort of special example of this kind of stress is fundraising.

So a scene from The Social Network. This is us partying and working at the same time—somebody's spraying champagne everywhere—The Social Network spends a lot of time painting these scenes. Mark's not in the scene, the other thing they spend all their time on is painting him out to be a huge jerk.

This is an actual scene from Palo Alto, he spent a lot of time at this desk, head down and focused. Mark was still kinda a jerk sometimes, but in this more like fun lovable way, and not in a sociopathic, scorned lover way. So this is just him signaling his intention to just be focused and keep working, not be social.

So then there's the scene demonstrating the insight moment, it's kind of like out of A Beautiful Mind, they literally stole that scene. So they like to paint that scene and jump to these moments from other moments, with partying in between. But really we were just at that table the whole time. So if you compare this photo, Mark is in the exact same position but he's wearing different clothes, so this is definitely a different day. That's what it's actually like in person. I just covered this bullet; this is the Economist article I was talking about a second ago.

So another form of stress is unwanted media attention. So part of it being glamorous is you get some positive media attention sometimes, it's nice to be on the cover of Time and to be the Person of the Year. It's maybe a little less nice to be on the cover of People with one of your wedding photos. It depends on who you are, I really hate it, but when Valleywag analyzes your lecture and tears you apart, you don't want that, you definitely don't want that. Nobody wants that.

One thing I almost never hear people talk about is you're much more committed. So if you're at a startup and it's very stressful and things are not going well, you're unhappy, you can just leave. For a founder, you can leave, but it's very uncool and pretty much a black eye for the rest of your career. And so you really are committed for ten years if it's going well and probably more like five years if it's not going well. So three years to figure out it's not going well and then if you find a nice landing for your company, another two years at the acquiring company. If you leave before that, again it's not only going to harm yourself financially but it's going to harm all your employees. So if you're lucky and you have a bad startup idea, you fail quickly, but most of the time it's not like that.

I should say, I've had a lot of this stress in my own life, especially in the early years of Facebook, I got really unhealthy, I wasn't exercising, I had a lot of anxiety actually throughout my back, like almost every six months, when I was twenty-one or twenty-two, which is pretty crazy. So if you do start a company, be aware that you're going to deal with this. You're going to have to actually manage this, it's one of your core responsibilities. Ben Horowitz likes to say the number one role of a CEO is managing your own psychology, it's absolutely true, make sure you do it.

Another reason, especially if you're had another job at another company, you start to develop this narrative, like the people running this company are idiots, they're making all these decisions and spending all their time in these stupid ways, I'm gonna start a company and I'm going to do it better. I'm going to set all the rules.

Sounds good, makes a lot of sense. If you've read my media post, you'll know what's coming, I'll give you guys a second to read this quote:

> People have this vision of being the CEO of a company they started and being on top of the pyramid. Some people are motivated by that, but that's not at all what it's like.
>
> What it's really like: everyone else is your boss – all of your employees, customers, partners, users, media are your boss. I've never had more bosses and needed to account for more people today.
>
> The life of most CEOs is reporting to everyone else, at least that's what it feels like to me and most CEOs I know. If you want to exercise power and authority over people, join the military or go into politics. Don't be an entrepreneur.
>
> *-Phil Libin*

This really resonates with me. One thing to point out is that the reality of these decision is nuanced. The people you thought were idiots probably weren't idiots, they just had a really difficult decision in front of them and people pulling them in multiple directions. So the most common thing I have to spend my time on and my energy on as a CEO is dealing with the problems that other people are bringing to me, the other priorities that people create, and it's usually in the form of a conflict. People want to go in different directions or customers want different things. And I might have my own opinions on that, but the game I'm playing is who do I disappoint the least and just trying to navigate all these difficult situations.

And even on a day to day basis, I might come in on Monday and have all these grand plans for how I'm going to improve the company. But if an important employee is threatening to quit, that's my number one priority. That's what I'm spending my time on.

A subset of You're the Boss is you have flexibility, you have control over your own schedule. This is a really attractive idea. So here's the reality:

> If you're going to be an entrepreneur, you will actually get some flex time to be honest. You'll be able to work any 24 hours a day you want!
>
> *-Phil Libin*

This truly resonates with me as well. Some of the reasons for this again, you're always on call. So maybe you don't intend to work all parts of the day, but you don't control which ones.

You're a role model of the company, and this is super important. So if you're an employee at a company, you might have some good weeks and you might have some bad weeks, some weeks when you're low energy and you might want to take a couple days off. That's really bad if you're an entrepreneur. Your team will really signal off of what you're bringing to the table. So if you take your foot off the gas, so will they.

You're always working anyways. If you're really passionate about an idea, it's going to pull you towards it. If you're working with great investors, you're working with great partners, they're going to be working really hard, they're going to want you to be working really hard.

Some companies like to tell the story about you can have your cake and eat it too, you can have like 4 days work weeks maybe, if you're Tim Ferris maybe you can have a 12 hours work week. It's a really attractive idea and it does work in a particular instance which is if you wanna actually have a small business to go after in each market then you are a small business entrepreneur, that makes little sense but as soon as you get past like 2 or 3 people you really need to step it up and be full-time committed.

You'll make more money and have more impact

This is the big one, the one I hear the most especially like candidates applying to Asana, they tell me "You know I'd really like to work for much smaller companies or start my own because then I have a much bigger slice of the pie or have much more impact on how that company does and I'll have more equity so I'll make more money as well". So let's examine when this might be true.

I'll explain these tables. They're a little complex but let's focus on the left first. These are just explaining Dropbox and Facebook, these are their current valuations and this is how much money you might make as employee number 100 coming into these companies especially if you're like an experienced, relatively experienced engineer, you have like 5 years of industry experience, you're pretty likely to have an offer that's around 10 base points. If you joined Dropbox couple years ago the upside you've already locked in is about $10M and there's plenty more growth from there. If you joined Facebook a couple years into its existence you've already made around $200M, this is a huge number and even if you joined Facebook as employee number 1000, so you joined like 2009, you still make $20M, that's a giant number and that's how you should be benchmarking when you're thinking about what you might make as an entrepreneur.

Moving over to the table on the right, these are two theoretical companies you might start. "Uber for Pet Sitting", pretty good idea if you're really well suited to this you might have a really good shot at building a $100M company and your share of that company is likely to be around 10%; that certainly fluctuates a lot, some founders have more than this, some founders have a lot less, but after multiple rounds of dilution, multiple rounds of option pool creation you're pretty likely to end up about here. If you have more than this I'd recommend Sam's post on equity split between founders and employees, you should be probably giving out more.

So basically if you're extremely confident in building a $100M, which is a big ask, it should go without saying that you should have a lot more confidence on Facebook in 2009 or Dropbox in 2014 that you might for a startup that doesn't even exist yet, then this is worth doing. If you have a $100M idea and you're pretty confident you can execute it I'd consider that.

If you think you're the right entrepreneur to build "Uber for Space Travel", that's a really huge idea, $2B idea, you're actually gonna have a pretty good return for that, you should definitely do that, this is also the value only after 4 years and this idea probably has legs, definitely go after that, if you're thinking of building that you probably shouldn't even be in this class right now, just go build that company.

So why is this financial reward and impact? I really think that financial reward is very strongly correlated with the impact we have on the world, if you don't believe that let's talk through some specific examples and not think about the equity at all.

So why might joining a late stage company actually might have a lot of impact, you get this force multiplier: they have an existing mass of user base, if it's Facebook it's a billion users, if it's Google it's a billion users, they have existing infrastructures you get to build on, that's also increasingly true for a new startup like AWS and all these awesome independent service providers, but you usually get some micro-proprietary technology and they maintain it for you, it's a pretty great place to start. And you get to work with a team, it'll help you leverage your ideas into something great.

So couple specific examples, Bret Taylor came into Google as around employee number 1500 and he invented Google Maps, that's a product you guys probably use everyday, I used it to get here and it's used by hundreds of millions of people around the world. He didn't need to start a company to do that, he happened to get a big financial reward, but the point is yet again massive impact.

My cofounder Justin Rosenstein joined Google a little later after Brett, he was a PM there and just as a side project he ended up prototyping a chat which used to be a stand-alone app, integrated in Gmail like you see in the upper right there and before he did that like you couldn't even think you could chat over Ajax or chat in the browser at all and he just kinda demonstrated it and showed it to his team and made it happen. This is probably a product most of you use almost everyday.

Perhaps even more impressively, shortly after that Justin left and became employee around 250 at Facebook and he led a hackaton project along with people like Andrew Bosworth and [?] to create the Like button, this is one of the most popular elements anywhere on the web, totally changed how people use it and then again didn't need to start a company to do it and almost certainly would have failed if he had tried because he really needed the distribution of Facebook to make it work.

So important to keep in mind the context for what kind of company you're trying to start and like where you will actually be able to make it happen.

So what's the best reason?

Sam already talked about this a little bit, but basically you can't not do it. You're super passionate about this idea, you're the right person to do it, you've gotta make it happen. So how does this break down?

This is a wordplay, you can't not do it in two ways. One is you're so passionate about it that you have to do it and you're going to do it anyways. This is really important because you'll need that passion to get through all of those hard parts of being an entrepreneur that we talked about earlier. You'll also need it to effectively recruit, candidates can smell when you don't have passion and there are enough entrepreneurs out there that do have passion so they may as well work for one of those! So this is table stakes for being an entrepreneur. Your subconscious can also tell when you don't have passion and that can be a huge problem.

The other way to interpret this is the world needs you to do it. This is validation that the idea is important, that it's going to make the world better, so the world needs it. If it's not something the world needs, go do something the world needs. Your time is really valuable, there are plenty of good ideas out there, maybe it's not your own, maybe it's at an existing company, but you may as well work on something that's going to be good.

The second way to interpret this is that the world needs you to do it. You're actually well suited for this problem in some way. If this isn't true, it may be a sign that your time is better spent somewhere else. But best case scenario if this isn't true, you outcompete the team for which it is true and it's a suboptimal outcome for the world and that doesn't feel very good.

So drawing this back to my own experience at Asana, Justin and I were reluctant entrepreneurs before we founded Asana, we were working at Facebook and we were working on a great problem. We would basically work all day long on our normal projects and then at night we would keep working on this internal task manager that was used internally at the company and it was just because we were so passionate about the idea, it was so clearly valuable that we couldn't do anything else.

And at some point we had to have the hard conversation of okay what does it mean if we don't actually start this company. We could see the impact it was having at Facebook, we were convinced it was valuable to the world. We were also convinced no one else was going to build it, the problem had been around a long time and we just kept seeing incremental solutions to it and so we believed if we didn't come out with the solution we thought was best, there would be a lot of value left on the table. We couldn't stop working on it and literally the idea was beating itself out of our chests and forcing itself out into the world. And I think that's really the feeling you should be looking for when you start a company, that's how you know you have the right idea.

I'll go ahead and stop there. I'll put some recommended books up here.

Thank you.

Lecture 2: Ideas, Products, Teams and Execution Part II

Sam Altman

Before I jump into today's lecture, I wanted to answer a few questions people had emailed me about the last lecture that we didn't have time for. So, if you have a question about what we covered last time, I am welcome to answer it now, starting with you.

Q: How do I identify if a market has a fast growth rate now and also for the next ten years?

A: The good news about this is this is one of the big advantages students have. You should just trust your instincts on this. Older people have to basically guess about the technologies young people are using. But you can just watch what you're doing and what your friends are doing and you will almost certainly have better instincts than anybody older than you. And so the answer to this is just trust your instincts, think about what you're doing more, think about what you're using, what you're seeing people your age using, that will almost certainly be the future.

Okay, one more question on the last lecture before we start.

Q: How do you deal with burnout while still being productive and remaining productive.

A: The answer to this is just that it sucks and you keep going. Unlike a student where you can throw up your hands and say you know I'm really burnt out and I'm just going to get bad grades this quarter, one of the hard parts about running a startup is that it's real life and you just have to get through it. The canonical advice is to go on a vacation and that never works for founders. It's sort of all consuming in this way that is very difficult to understand.

So what you do is you just keep going. You rely on people, it's really important, founder depression is a serious thing and you need to have a support network. But the way through burn out is just to address the challenges, to address the things that are going wrong and you'll eventually feel better.

Last lecture, we covered the idea and the product and I want to emphasize that if you don't get those right, none of the rest of this is going to save you. Today, we're going to talk about how to hire and how to execute. Hopefully you don't execute the people you hire. Sometimes.

First, I want to talk about cofounders. Cofounder relationships are among the most important in the entire company. Everyone says you have to watch out for tension brewing among cofounders and you have to address is immediately. That's all true and certainly in YC's case, the number one cause of early death for startups is cofounder blowups. But for some reason, a lot of people treat choosing their cofounder with even less importance than hiring. Don't do this! This is one of the most important decisions you make in the life of your startup and you need to treat it as such.

And for some reason, students are really bad at this. They just pick someone. They're like, I want to start a business and you want to start a business, let's start a startup together. There are these cofounder dating things where you're like, Hey I'm looking for a cofounder, we don't really know each other, let's start a company. And this is like, crazy. You would never hire someone like this and yet people are willing to choose their business partners this way. It's really really bad. And choosing a random random cofounder, or choosing someone you don't have a long history with, choosing someone you're not friends with, so when things are really going wrong, you have this sort of past history to bind you together, usually ends up in disaster.

We had one YC batch in which nine out of about seventy-five companies added on a new cofounder between when we interviewed the companies and when they started, and all nine of those teams fell apart within the next year. The track record for companies where the cofounders don't know each other is really bad.

A good way to meet a cofounder is to meet in college. If you're not in college and you don't know a cofounder, the next best thing I think is to go work at an interesting company. If you work at Facebook or Google or something like that, it's almost as cofounder rich as Stanford. It's better to have no cofounder than to have a bad cofounder, but it's still bad to be a solo founder. I was just looking at the stats here before we started. For the top, and I may have missed one because I was counting quickly, but I think, for the top twenty most valuable YC companies, almost all of them have at least two founders. And we probably funded a rate of like one out of ten solo teams.

So, best of all, cofounder you know, not as good as that, but still okay, solo founder. Random founder you meet, and yet students do this for some reason, really really bad.

So as you're thinking about cofounders and people that could be good, there's a question of what you're looking for right? At YC we have this public phrase, and it's relentlessly resourceful, and everyone's heard of it. And you definitely need relentlessly resourceful cofounders, but there's a more colorful example that we share at the YC kickoff. Paul Graham started using this and I've kept it going.

So, you're looking for cofounders that need to be unflappable, tough, they know what to do in every situation. They act quickly, they're decisive, they're creative, they're ready for anything, and it turns out that there's a model for this in pop culture. And it sounds very dumb, but it's at least very memorable and we've told every class of YC this for a long time and I think it helps them.

And that model is James Bond. And again, this sounds crazy, but it will at least stick in your memory and you need someone that behaves like James Bond more than you need someone that is an expert in some particular domain.

As I mentioned earlier, you really want to know your cofounders for awhile, ideally years. This is especially true for early hires as well, but incidentally, more people get this right for early hires than they do for cofounders. So, take advantage of school. In addition to relentlessly resourceful, you want a tough and a calm cofounder. There are obvious things like smart, but everyone knows you want a smart cofounder, they don't prioritize things like tough and calm enough, especially if you feel like you yourself aren't, you need a cofounder who is. If you aren't technical, and even if most of the people in this room feel like they are, you want a technical cofounder. There's this weird thing going on in startups right now where it's become popular to say, You know what, we don't need a technical cofounders, we're gonna hire people, we're just gonna be great managers.

That doesn't work too well in our experience. Software people should really be starting software companies. Media people should be starting media companies. In the YC experience, two or three cofounders seems to be about perfect. One, obviously not great, five, really bad. Four works sometimes, but two or three I think is the target.

The second part of how to hire: try not to. One of the weird things you'll notice as you start a company, is that everyone will ask you how many employees you have. And this is the metric people use to judge how real your startup is and how cool you are. And if you say you have a high number of employees, they're really impressed. And if you say you have a low number of employees, then you sound like this little joke. But actually it sucks to have a lot of employees, and you should be proud of how few employees you have. Lots of employees ends up with things like a high burn rate, meaning you're losing a lot of money every month, complexity, slow decision making, the list goes on and it's nothing good.

So you want to be proud of how much you can get done with a small numbers of employees. Many of the best YC companies have had a phenomenally small number of employees for their first year, sometimes none besides the founders. They really try to stay small as long as they possibly can. At the beginning, you should only hire when you desperately need to. Later, you should learn to hire fast and scale up the company, but in the early days the goal should be not to hire. And one of the reasons this is so bad, is that the cost of getting an early hire wrong is really high. In fact, a lot of the companies that I've been very involved with, that have had a very bad early hire in the first three or so employees never recover, it just kills the company.

Airbnb spent five months interviewing their first employee. And in their first year, they only hired two. Before they hired a single person, they wrote down a list of the culture values that they wanted any Airbnb employee to have. One of those what that you had to bleed Airbnb, and if you didn't agree to that they just wouldn't hire you. As an example of how intense Brian Chesky is, he's the Airbnb CEO, he used to ask people if they would take the job if they got a medical diagnosis that they have one year left to life. Later he decided that that was a little bit too crazy and I think he relaxed it to ten years, but last I heard, he still asks that question.

These hires really matter, these people are what go on to define your company, and so you need people that believe in it almost as much as you do. And it sounds like a crazy thing to ask, but he's gotten this culture of extremely dedicated people that come together when the company faces a crisis. And when the company faced a big crisis early on, everyone lived in the office, and they shipped product every day until the crisis was over. One of the remarkable observations about Airbnb is that if you talk to any of the first forty or so employees, they all feel like they were a part of the founding of the company.

But by having an extremely high bar, by hiring slowly ensures that everyone believes in the mission, you can get that. So let's say, you listened to the warning about not hiring unless you absolutely have too. When you're in this hiring mode, it should be your number one priority to get the best people. Just like when you're in product mode that should be your number one priority. And when you're in fundraising mode, fundraising is your number one priority.

On thing that founders always underestimate is how hard it is to recruit. You think you have this great idea and everyone's going to join. But that's not how it works. To get the very best people, they have a lot of great options and so it can easily take a year to recruit someone. It's this long process and so you have to convince them that your mission is the most important of anything that they're looking at. This is another case of why it's really important to get the product right before looking at anything else. The best people know that they should join a rocketship.

By the way, that's my number one piece of advice if you're going to join a startup, is pick a rocketship. Pick a company that's already working and that not everyone yet realizes that, but you know because you're paying attention, that it's going to be huge. And again, you can usually identify these. But good people know this, and so good people will wait, to see that you're on this trajectory before they join.

One question that people asked online this morning was how much time you should be spending on hiring. The answer is zero or twenty-five percent. You're either not hiring at all or it's probably your single biggest block of time. In practice, all these books on management say you should spend fifty percent of your time hiring, but the people that give that advice, it's rare for them to even spend ten percent themselves. Twenty-five percent is still a huge amount of time, but that's really how much you should be doing once you're in hiring mode.

If you compromise and hire someone mediocre you will always regret it. We like to warn founders of this but no one really feels it until they make the mistake the first time, but it can poison the culture. Mediocre people at huge companies will cause some problems, but it won't kill the company. A single mediocre hire within the first five will often in fact kill a startup.

A friend of mine has a sign up in the conference room that he uses for interviews and he positions the sign that the candidate is looking at it during the interview and it says that mediocre engineers do not build great companies. Yeah that's true, it’s really true. You can get away with it in a big company because people just sort of fall through the cracks but every person at a startup sets the tone. So if you compromise in the first five, ten hires it might kill the company. And you can think about that for everyone you hire: will I bet the future of this company on this single hire? And that's a tough bar. At some point in the company, when you're bigger, you will compromise on a hire. There will be some pressing deadline or something like that you will still regret. But this is the difference between theory and practice we're going to have later speakers talk about what to do when this happens. But in the early days you just can't screw it up.

Sources of candidates. This is another thing that students get wrong a lot. The best source for hiring by far is people that you already know and people that other employees in the company already know. Most great companies in text have been built by personal referrals for the first hundred employees and often many more. Most founders feel awkward but calling anyone good that they've ever met and asking their employees to do the same. But she'll notice if you go to work at Facebook or Google one of the things they do in your first few weeks is an HR person sits you down and beat out of you every smart person you’ve ever met to be able to recruit them.

These personal referrals really are the trick to hiring. Another tip is to look outside the valley. It is brutally competitive to hire engineers here but you probably know people elsewhere in the world that would like to work with you.

Another question that founders ask us a lot about his experience and how much that matters. The short version here is that experience matters for some roles and not for others. When you're hiring someone that is going to run a large part of your organization experience probably matters a lot. For most of the early hires that you make at a startup, experience probably doesn't matter that much and you should go for aptitude and belief in what you're doing. Most of the best hires that I've made in my entire life have never done that thing before. So it's really worth thinking, is this a role where I care about experience or not. And you'll often find to don't, especially in the early days.

There are three things I look for in a hire. Are they smart? Do they get things done? Do I want to spend a lot of time around them? And if I get an answer, if I can say yes to all three of these, I never regret it, it's almost always worked out. You can learn a lot about all three of these things in an interview but the very best way is working together, so ideally someone you've worked together with in the past and in that case you probably don't even need an interview. If you haven't, then I think it's way better to work with someone on a project for a day or two before hiring them. You'll both learn a lot they will too and most first-time founders are very bad interviewers but very good at evaluating someone after they've worked together.

So one of the pieces of advice that we give at YC is try to work on a project together instead of an interview. If you are going to interview, which you probably will, you should ask specifically about projects that someone worked on in the past. You'll learn a lot more than you will with brainteasers. For some reason, young technical cofounders love to ask brainteasers rather than just ask what someone has done. Really dig in to projects people have worked on. And call references. That is another thing that first time founders like to skip. You want to call some people that these people have worked with in the past. And when you do, you don't just want to ask, How was so-and-so, you really want to dig in. Is this person in the top five percent of people you've ever worked with? What specifically did they do? Would you hire them again? Why aren't you trying to hire them again? You really have to press on these reference calls.

Another thing that I have noticed from talking to YC companies is that good communication skills tend to correlate with hires that work out. I used to not pay attention to this. We're going to talk more about why communication is so important in an early startup. If someone is difficult to talk to, if someone cannot communicate clearly, it's a real problem in terms of their likelihood to work out. Also. for early employees you want someone that has somewhat of a risk-taking attitude. You generally get this, otherwise they wouldn't be interested in a startup, but now that startups are sort of more in fashion, you want people that actually sort of like a little bit of risk. If someone is choosing between joining McKinsey or your startup it's very unlikely they're going to work out at the startup.

You also want people who are maniacally determined and that is slightly different than having a risk tolerant attitude. So you really should be looking for both. By the way, people are welcome to interrupt me with questions as stuff comes up.

There is a famous test from Paul Graham called the animal test. The idea here is that you should be able to describe any employee as an animal at what they do. I don't think that translates out of English very well but you need unstoppable people. You want people that are just going to get it done. Founders who usually end up being very happy with their early hires usually end up describing these people as the very best in the world at what they do.

Mark Zuckerberg once said that he tries to hire people that A. he'd be comfortable hanging with socially and B. he'd be comfortable reporting to if the roles were reversed. This strikes me as a very good framework. You don't have to be friends with everybody, but you should at least enjoy working with them. And if you don't have that, you should at least deeply respect them. But again, if you don't want to spend a lot of time around people you should trust your instincts about that.

While I'm on this topic of hiring, I want to talk about employee equity. Founders screw this up all the time. I think as a rough estimate, you should aim to give about ten percent of the company to the first ten employees.

They have to earn it over four years anyway, and if they're successful, they're going to contribute way more than that. They're going to increase the value of the company way more than that, and if they don't then they won't be around anyway.

For whatever reason founders are usually very stingy with equity to employees and very generous with equity for investors. I think this is totally backwards. I think this is one of the things founders screw up the most often. Employees will only add more value over time. Investors will usually write the check and then, despite a lot of promises, don't usually do that much. Sometimes they do, but your employees are really the ones that build the company over years and years.

So I believe in fighting with investors to reduce the amount of equity they get and then being as generous as you possibly can with employees. The YC companies that have done this well, the YC companies that have been super generous with their equity to early employees, in general, are the most successful ones that we've funded.

One thing that founders forget is that after they hire employees, they have to retain them. I'm not going to go into full detail here because we're going to have a lecture on this later, but I do want to talk about it a little bit because founders get this wrong so often. You have to make sure your employees are happy and feel valued. This is one of the reasons that equity grants are so important. People in the excitement of joining a startup don't think about it much, but as they come in day after day, year after year, if they feel they have been treated unfairly that will really start to grate on them and resentment will build.

But more than that, learning just a little bit of management skills, which first-time CEOs are usually terrible at, goes a long way. One of the speakers at YC this summer, who is now extremely successful, struggled early on and had his team turn over a few times. Someone asked him what his biggest struggle was and he said, turns out you shouldn't tell your employees they're fucking up every day unless you want them all to leave because they will.

But as a founder, this is a very natural instinct. You think you can do everything the best and it’s easy to tell people when they’re not doing it well. So learning just a little bit here will prevent this massive team churn. It also doesn't come naturally to most founders to really praise their team. It took me a little while to learn this too. You have to let your team take credit for all the good stuff that happens, and you take responsibility for the bad stuff.

You have to not micromanage. You have to continually give people small areas of responsibility. These are not the things that founders think about. I think the best thing you can do as a first-time founder is to be aware that you will be a very bad manager and try to overcompensate for that. Dan Pink talks about these three things that motivate people to do great work: autonomy, mastery, and purpose. I never thought about that when I was running my company but I've thought about since and I think that’s actually right. I think it's worth trying to think about that. It also took me a while to learn to do things like one on one and to give clear feedback.

All of these things are things first time CEO don't normally do, and maybe I can save you from not doing that.

The last part on the team section is about firing people when it's not working. No matter what I say here is not going to prevent anyone from doing it wrong and the reason that I say that is that firing people is one of the worst parts of running a company. Actually in my own experience, I'd say it is the very worst part. Every first time founder waits too long, everyone hopes that an employee will turn around. But the right answer is to fire fast when it's not working. It's better for the company, it's also better for the employee. But it's so painful and so awful, that everyone gets it wrong the first few times.

In addition to firing people who are doing bad at their job, you also wanna fire people who are a) creating office politics, and b) who are persistently negative. The rest of the company is always aware of employees doing things like this, and it's just this huge drag - it's completely toxic to the company. Again, this is an example of something that might work OK in a big company, although I'm still skeptical, but will kill a startup. So that you need to watch out for people that are ifs.

So, the question is, how do you balance firing people fast and making early employees feel secure? The answer is that when an employee's not working, it's not like they screw up once or twice. Anyone will screw up once or twice, or more times than that, and you know you should be like very loving, not take it out on them, like, be a team, work together.

If someone is getting every decision wrong, that's when you need to act, and at that point it'll be painfully aware to everyone. It's not a case of a few screw-ups, it's a case where every time someone does something, you would have done the opposite yourself. You don't get to make their decisions but you do get to choose the decision-makers. And, if someone's doing everything wrong, just like a consistent thing over like a period of many weeks or a month, you'll be aware of it.

This is one of those cases where in theory, it sounds complicated to be sure what you're talking about, and in practice there's almost never any doubt. It's the difference between someone making one or two mistakes and just constantly screwing everything up, or causing problems, or making everyone unhappy, is painfully obvious the first time you see it.

When should co-founders decide on the equity split?

For some reason, I've never really been sure why this is, a lot of founders, a lot of co-founders like to leave this off for a very long time. You know, they'll even sign the incorporation documents in some crazy way so that they can wait to have this discussion.

This is not a discussion that gets easier with time, you wanna set this ideally very soon after you start working together. And it should be near-equal. If you're not willing to give someone - your co-founder - you know, like an equal share of the equity, I think that should make you think hard about whether or not you want them as a co-founder. But in any case, you should try to have the ink dry on this before the company gets too far along. Like, certainly in the first number of weeks.

So the question is - I said that inexperience is OK - how do you know if someone's gonna scale past, not scale up to a role, as things go on and later become crippling. People that are really smart and that can learn new things can almost always find a role in the company as time goes on. You may have to move them into something else, something other than where they started. You know, it may be that you hire someone to lead the engineering team that over time can't scale as you get up to 50 people, and you give them a different role. Really good people that can almost find some great place in the company, I have not seen that be a problem too often.

So the question is what happens when your relationship with your cofounder falls apart. We're gonna have a session on mechanics later on in the course, but here is the most important thing that founders screw up. Which is, every cofounder, you yourself of course, has to have vesting. Basically what you're doing with cofounder vesting is you're pre-negotiating what happens if one of you leaves. And so the normal stance on this in Silicon Valley is that it takes four years, let's say you split the equity fifty-fifty, is that it takes four years to earn all of that. And the clock doesn't start until one year in. So if you leave after one year, you keep twenty-five percent of the equity, and if you leave after two years, fifty, and on and on like that.

If you don't do that and if you have a huge fallout and one founder leaves early on with half the company, you have this deadweight on your equity table, and it's very hard to get investors to fund you or to do anything else. So number one piece of advice to prevent that is to have vesting on the equity. We pretty much won't fund a company now where the founders don't have vested equity because it's just that bad. The other thing that comes up in the relationship between the cofounders, which happens to some degree in every company, is talk about it early, don't let it sit there and fester.

If you have to choose between hiring a sub-optimal employee and losing your customers to a competitor, what do you do? If it's going to be one of the first five employees at a company I would lose those customers. The damage that it does to the company- it's better to lose some customers than to kill the company. Later on, I might have a slightly different opinion, but it's really hard to say in the general case.

I am going to get to that later. The question is: what about cofounders that aren't working in the same location? The answer is, don't do it. I am skeptical of remote teams in general but in the early days of a startup, when communication and speed outweigh everything else, for some reason video conferencing calls just don't work that well. The data on this is look at say the 30 successful software companies of all time and try to point to a single example where the cofounders were in different locations. It's really really tough.

Alright, so now we're going to talk about execution. Execution for most founders is not the most fun part of running the company, but it is the most critical. Many cofounders think they're just signing up to this beautiful idea and then they're going to go be on magazine covers and go to parties. But really what it’s about more than anything else, what being a cofounder really means, is signing up for this years long grind on execution and you can’t outsource this.

The way to have a company that executes well is you have to execute well yourself. Every thing at a startup gets modeled after the founders. Whatever the founders do becomes the culture. So if you want a culture where people work hard, pay attention to detail, manage the customers, are frugal, you have to do it yourself. There is no other way. You cannot hire a COO to do that while you go off to conferences. The company just needs to see you as this maniacal execution machine. As I said in the first lecture, there's at least a hundred times more people with great ideas than people who are willing to put in the effort to execute them well. Ideas by themselves are not worth anything, only executing well is what adds and creates value.

A big part of execution is just putting in the effort, but there is a lot you can learn about how to be good at it. And so we're going to have three classes that just talk about this.

The CEO, people ask me all the time about the jobs of the CEO. There are probably more than five, here are five that come up a lot in the early days. The first four everyone thinks of as CEO jobs: set the vision, raise money, evangelize the mission to people you're trying to recruit, executives, partners, press, everybody, hire and manage the team. But the fifth one is setting the execution bar and this is not the one that most founders get excited about or envision themselves doing but I think it is actually one of the critical CEO roles and no one but the CEO can do this.

Execution gets divided into two key questions. One, can you figure out what to do and two, can you get it done. So I want to talk about two parts of getting it done, assuming that you've already figured out what to do. And those are focus and intensity. So focus is critical. One of my favorite questions to ask founders about what they're spending their time and their money on. This reveals almost everything about what founders think is important.

One of the hardest parts about being a founder is that there are a hundred important things competing for your attention every day. And you have to identify the right two or three, work on those, and then ignore, delegate, or defer the rest. And a lot of these things that founders think are important, interviewing a lot at different law firms, going to conferences, recruiting advisers, whatever, they just don't matter. What really does matter varies with time, but it's an important piece of advice. You need to figure out what the one or two most important things are, and then just do those.

And you can only have two or three things every day, because everything else will just come at you. There will be fires every day and if you don't get good at setting what those two or three things are, you'll never be good at getting stuff done. This is really hard for founders. Founders get excited about starting new things.

Unfortunately the trick to great execution is to say no a lot. You're saying no ninety-seven times out of a hundred, and most founders find they have to make a very conscious effort to do this. Most startups are nowhere near focused enough. They work really hard-maybe-but they don't work really hard at the right things, so they'll still fail. One of the great and terrible things about starting a start up is that you get no credit for trying. You only get points when you make something the market wants. So if you work really hard on the wrong things, no one will care.

So then there's this question of how do you figure out what to focus on each day. Each day it's really important to have goals. Most good founders I know have a set of small overarching goals for the company that everybody in the company knows. You know it could be something like ship a product by this date, get this certain growth rate, get this engagement rate, hire for these key roles, those are some of them but everyone in the company can tell you each week what are our key goals. And then everybody executes based off of that.

The founders really set the focus. Whatever the founders care about, whatever the founders focus on, that's going to set the goals for the whole company. The best founders repeat these goals over and over, far more often than they think they should need to. They put them up on the walls they talk about them in one on ones and at all-hands meetings each week. And it keeps the company focus. One of the keys to focus, and why I said cofounders that aren't friends really struggle, is that you can't be focused without good communication. Even if you have only four or five people at a company, a small communication breakdown is enough for people to be working on slightly different things. And then you lose focus and the company just scrambles.

I'm going to talk about this a little bit later, but growth and momentum are something you can never lose focus on. Growth and momentum are what a startup lives on and you always have to focus on maintaining these. You should always know how you're doing against your metrics, you should have a weekly review meeting every week, and you should be extremely suspicious if you’re ever talking about, we’re not focused on growth right now, we’re not growing that well right now but we're doing this other thing, we don't have a timeline for when we are going to ship this because we're focused on this other thing, we’re doing a re-brand, whatever, almost always a disaster.

So you want to have the right metrics and you want to be focused on growing those metrics and having momentum. Don't let the company get distracted or excited about other things. A common mistake is that companies get excited by their own PR. It's really easy to get PR with no results and it actually feels like you're really cool. But in a year you'll have nothing, and at that point you won't be cool anymore, and you'll just be talking about these articles from a year ago that, Oh you know these Stanford students start a new start up, it's going to be the next big thing and now you have nothing and that sucks.

As I mentioned already, be in the same space. I think this is pretty much a nonstarter. Remote confounding teams is just really really hard. It slows down the cycle time more than anybody ever thinks it's going to.

The other piece besides focus for execution is intensity. Startups only work at a fairly intense level. A friend of mine says the secret to start up success is extreme focus and extreme dedication. You can have a startup and one other thing, you can have a family, but you probably can't have many other things. Startups are not the best choice for work life balance and that's sort of just the sad reality. There's a lot of great things about a startup, but this is not one of them. Startups are all-consuming in a way that is generally difficult to explain. You basically need to be willing to outwork your competitors.

The good news here is that a small amount of extra work on the right thing makes a huge difference. One example that I like to give is thinking about the viral coefficient for a consumer web product. How many new users each existing user brings in. If it's .99 the company will eventually flatline and die. And if it's 1.01 you'll be in this happy place of exponential growth forever.

So this is one concrete example of where a tiny extra bit of work is the difference between success and failure. When we talk to successful founders they tell stories like this all the time. Just outworking their competitors by a little bit was what made them successful.

So you have to be really intense. This only comes from the CEO, this only comes from the founders. One of the biggest advantages that start ups have is execution speed and you have to have this relentless operating rhythm. Facebook has this famous poster that says move fast and break things. But at the same time they manage to be obsessed with quality. And this is why it's hard. It's easy to move fast or be obsessed with quality, but the trick is to do both at a startup. You need to have a culture where the company has really high standards for everything everyone does, but you still move quickly.

Apple, Google, and Facebook have each done this extremely well. It's not about the product, it's about everything they do. They move fast and they break things, they're frugal in the right places, but they care about quality everywhere. You don't buy people shitty computers if you don't want them to write shitty code. You have to set a quality bar that runs through the entire company. Related to this is that you have to be decisive. Indecisiveness is a startup killer. Mediocre founders spend a lot of time talking about grand plans, but they never make a decision. They're talking about you know I could do this thing, or I could do that other thing, and they're going back and forth and they never act. And what you actually need is this bias towards action.

The best founders work on things that seem small but they move really quickly. But they get things done really quickly. Every time you talk to the best founders they've gotten new things done. In fact, this is the one thing that we learned best predicts a success of founders in YC. If every time we talk to a team they've gotten new things done, that's the best predictor we have that a company will be successful. Part of this is that you can do huge things in incremental pieces. If you keep knocking down small chunks one at a time, in a year you look back and you've done this amazing thing. On the other hand, if you disappear for a year and you expect to come back with something amazing all at once, it usually never happens.

So you have to pick these right size projects. Even if you're building this crazy synthetic biology company and you say well I have to go away for a year, there's no way to do this incrementally, you can still usually break it into smaller projects.

So speed is this huge premium. The best founders usually respond to e-mail the most quickly, make decisions most quickly, they're generally quick in all of these ways. And they had this do what ever it takes attitude.

They also show up a lot.

They come to meetings, they come in, they meet us in person. One piece of advice that I have that's always worked for me: they get on planes in marginal situations. I'll tell a quick story here.

When I was running my own company, we found out we were about to lose a deal. It was sort of this critical deal from the first big customer in the space. And it was going to go to this company that had been around for year before we were. And they had this like all locked up. And we called and said "we have this better product you have to meet with us" and they said "well we're signing this deal tomorrow. sorry." We drove to the airport, we got on a plane, we were at their office at 6am the next morning. We just sat there, they told us to go away, we just kept sitting there. Finally once of the junior guys decided to meet with us, after that, finally one of the senior guys decided to meet with us. They ended up ripping up the contract with the other company, and we closed the deal with them about a week later. And I'm sure, that had we not gotten on a plane, had we not shown up in person, that would not have worked out.

And so, you just sort of show and and do these things, when people say get on plane in marginal situations, they actually mean it, but they don't mean it literally. But I actually think it's good, literal advice.

So I mentioned this momentum and growth earlier. Once more: the momentum and growth are the lifeblood of startups. This is probably in the top three secrets of executing well. You want a company to be winning all the time. If you ever take your foot off the gas pedal, things will spiral out of control, snowball downwards. A winning team feels good and keeps winning. A team that hasn't won in a while gets demotivated and keeps losing. So always keep momentum, it's this prime directive for managing a startup. If I can only tell founders one thing about how to run a company, it would be this.

For most software startups, this translates to keep growing. For hardware startups it translates to: don't let your ship dates slip. This is what we tell people during YC, and they usually listen and everything is good. What happens at the end of YC is that they get distracted on other things, and then growth slows down. And somehow, after that happens, people start getting unhappy and quitting and everything falls apart. It's hard to figure out a growth engine because most companies grow in new ways, but there's this thing: if you build a good product it will grow. So getting this product right at the beginning is the best way not to lose momentum later.

If you do lose momentum, most founders try to get it back in the wrong way. They give these long speeches about vision for the company and try to rally the troops with speeches. But employees in a company where momentum has sagged, don't want to hear that. You have to save the vision speeches for when the company is winning. When you're not winning, you just have to get momentum back in small wins. A board member of mine used to say that sales fix everything in a startup. And that is really true. So you figure out where you can get these small wins and you get that done. And then you'll be amazed at how all the other problems in a startup disappear.

Another thing that you'll notice if you have momentum sag, is that everyone starts disagreeing about what to do. Fights come out when a company loses momentum. And so a framework for that that I think works is that when there's disagreement among the team about what to do, then you ask your users and you do whatever your users tell you. And you have to remind people: "hey, stuff's not working right now we don't actually hate each other, we just need to get back on track and everything will work." If you just call it out, if you just acknowledge that, you'll find that things get way better.

To use a Facebook example again, when Facebook's growth slowed in 2008, mark instituted a "growth group." They worked on very small things to make Facebook grow faster. All of these by themselves seemed really small, but they got the curve of Facebook back up. It quickly became the most prestigious group there. Mark has said that it's been one of Facebook's best innovations. According to friends of mine that worked at Facebook at the time, it really turned around the dynamic of the company. And it went from this thing where everyone was feeling bad, and momentum was gone, back to a place that was winning.

So a good way to keep momentum is to establish an operating rhythm at the company early. Where you ship product and launch new features on a regular basis. Where you're reviewing metrics every week with the entire company. This is actually one of the best things your board can do for you. Boards add value to business strategy only rarely. But very frequently you can use them as a forcing function to get the company to care about metrics and milestones.

One thing that often disrupts momentum and really shouldn't is competitors. Competitors making noise in the press I think probably crushes a company's momentum more often than any other external factor.

So here's a good rule of thumb: don't worry about a competitor at all, until they're actually beating you with a real, shipped product. Press releases are easier to write than code, and that is still easier than making a great product. So remind your company of this, and this is sort of a founder's role, is not to let the company get down because of the competitors in the press.

This great quote from Henry Ford that I love: "The competitor to be feared is one who never bothers about you at all, but goes on making his own business better all the time."

These are almost never the companies that put out a lot of press releases. And they bum people out.

Lecture 3: Counterintuitive Parts of Startups, and How to Have Ideas

Paul Graham

Paul Graham Ft: Ben Yu Produced By: Stanford & Y Combinator Verified Annotations By: Ben Yu PYONG! 21 FacebookTwitter Embed Follow This text is annotated! Click on the highlights to read what others are saying. If you'd like to add your own insights, comments, or questions to specific parts of the lecture, visit the lecture page on Genius, highlight the relevant text, and click the button that pops up. Your annotation will appear both here and on Genius.

One of the advantages of having kids is that when you have to give advice to people you can ask yourself, "what would I tell my own kids?", and actually you'll find this really focuses you. So even though my kids are little, my two year old today, when asked what he'll be after two, said "a bat." The correct answer was three, but "a bat" is so much more interesting. So even though my kids are little, I already know what I would tell them about startups, if they were in college, so that is what I'm going to tell you. You're literally going to get what I would tell my own kids, since most of you are young enough to be my own kids.

Startups are very counterintuitive and I'm not sure exactly why. It could be simply because knowledge about them has not permeated our culture yet, but whatever the reason, this is an area where you cannot trust your intuition all the time. It's like skiing in that way - any of you guys learn to ski as adults? When you first try skiing and you want to slow down, your first impulse is to lean back, just like in everything else. But lean back on the skis and you fly down the hill out of control. So, as I learned, part of learning to ski is learning to suppress that impulse. Eventually you get new habits, but in the beginning there is this list of things you're trying to remember as you start down the hill: alternate feet, make s-turns, do not drag the inside foot, all this stuff.

Startups are as unnatural as skiing and there is a similar list of stuff you have to remember for startups. What I'm going to give you today is the beginning of the list, the list of the counterintuitive stuff you have to remember to prevent your existing instincts from leading you astray.

The first thing on it is the fact I just mentioned: startups are so weird that if you follow your instincts they will lead you astray. If you remember nothing more than that, when you're about to make a mistake, you can pause before making it. When I was running Y Combinator we used to joke that our function was to tell founders things they would ignore, and it's really true. Batch after batch the YC partners warned founders about mistakes they were about to make and the founders ignored them, and they came back a year later and said, "I wish we'd listened." But that dude is in their cap table and there is nothing they can do.

Q: Why do founders persistently ignore the partner’s advice?

A: That's the thing about counterintuitive ideas, they contradict your intuitions, they seem wrong, so of course your first impulse is to ignore them and, in fact, that's not just the curse of Y Combinator, but to some extent our raison d'être. You don't need people to give you advice that does not surprise you. If founders' existing intuition gave them the right answers, they would not need us. That's why there are a lot of ski instructors, and not many running instructors; you don't see those words together, "running instructor," as much as you see "ski instructor." It's because skiing is counterintuitive, sort of what YC is—business ski instructors—except you are going up slopes instead of down them, well ideally.

You can, however, trust your instincts about people. Your life so far hasn't been much like starting a startup, but all the interactions you've had with people are just like the interactions you have with people in the business world. In fact, one of the big mistakes that founders make is to not trust their intuition about people enough. They meet someone, who seems impressive, but about whom they feel some misgivings and then later when things blow up, they say, "You know I knew there was something wrong about that guy, but I ignored it because he seemed so impressive."

There is this specific sub-case in business, especially if you come from an engineering background, as I believe you all do. You think business is supposed to be this slightly distasteful thing. So when you meet people who seem smart, but somehow distasteful, you think, "Okay this must be normal for business," but it's not. Just pick people the way you would pick people if you were picking friends. This is one of those rare cases where it works to be self indulgent. Work with people you would generally like and respect and that you have known long enough to be sure about because there are a lot of people who are really good at seeming likable for a while. Just wait till your interests are opposed and then you'll see.

The second counterintuitive point, this might come as a little bit of a disappointment, but what you need to succeed in a startup is not expertise in startups. That makes this class different from most other classes you take. You take a French class, at the end of it you've learned how to speech French. You do the work, you may not sound exactly like a French person, but pretty close, right? This class can teach you about startups, but that is not what you need to know. What you need to know to succeed in a startup is not expertise in startups, what you need is expertise in your own users.

Mark Zuckerberg did not succeed at Facebook because he was an expert in startups, he succeeded despite being a complete noob at startups; I mean Facebook was first incorporated as a Florida LLC. Even you guys know better than that. He succeeded despite being a complete noob at startups because he understood his users very well. Most of you don't know the mechanics of raising an angel round, right? If you feel bad about that, don't, because I can tell you Mark Zuckerberg probably doesn't know the mechanics of raising an angel round either; if he was even paying attention when Ron Conway wrote him the big check, he probably has forgotten about it by now.

In fact, I worry it's not merely unnecessary for people to learn in detail about the mechanics of starting a startup, but possibly somewhat dangerous because another characteristic mistake of young founders starting startups is to go through the motions of starting a startup. They come up with some plausible sounding idea, they raise funding to get a nice valuation, then the next step is they rent a nice office in SoMa and hire a bunch of their friends, until they gradually realize how completely fucked they are because while imitating all the outward forms of starting a startup, they have neglected the one thing that is actually essential, which is to make something people want. By the way that's the only use of that swear word, except for the initial one, that was involuntary and I did check with Sam if it would be okay; he said he had done it several times, I mean use the word.

We saw this happen so often, people going through the motion of starting a startup, that we made up a name for it: "Playing House." Eventually I realized why it was happening, the reason young founders go though the motions of starting a startup is because that is what they have been trained to do, their whole life, up to this point. Think about what it takes to get into college: extracurricular activities? Check. Even in college classes most of the work you do is as artificial as running laps, and I'm not attacking the educational system for being this way, inevitably the work that you do to learn something is going to have some amount of fakeness to it. And if you measure people's performance they will inevitably exploit the difference to the degree that what you're measuring is largely an artifact of the fakeness.

I confess that I did this myself in college; in fact, here is a useful tip on getting good grades. I found that in a lot of classes there might only be twenty or thirty ideas that had the right shape to make good exam questions. So the way I studied for exams in these classes was not to master the material in the class, but to try and figure out what the exam questions would be and work out the answers in advance. For me the test was not like, what my answers would be on my exam, for me the test was which of my exam questions would show up on the exam. So I would get my grade instantly, I would walk into the exam and look at the questions and see how many I got right, essentially. It works in a lot of classes, especially CS classes. I remember automata theory, there are only a few things that make sense to ask about automata theory.

So it's not surprising that after being effectively trained for their whole lives to play such games, young founders' first impulse on starting a startup is to find out what the tricks are for this new game. What are the extracurricular activities of startups, what are things I have to do? They always want to know, since apparently the measure of success for a startup is fundraising, another noob mistake. They always want to know, what are the tricks for convincing investors? And we have to tell them the best way to convince investors is to start a startup that is actually doing well, meaning growing fast, and then simply tell investors so.

Then they ask okay, so what are the tricks for growing fast, and this is exacerbated by the existence of this term, "Growth

Hacks." Whenever you hear somebody talk about Growth Hacks, just mentally translate it in your mind to "bullshit," because what we tell them is the way to make your startup grow is to make something that users really love, and then tell them about it. So that's what you have to do: that's Growth Hacks right there.

So many of the conversations the YC partners have with the founders begin with the founders saying a sentence that begins with, "How do I," and the partners answering with a sentence that begins with, "Just." Why do they make things so complicated? The reason, I realized, after years of being puzzled by this, is they're looking for the trick, they've been trained to look for the trick.

So, this is the third counterintuitive thing to remember about startups: starting a startup is where gaming the system stops working. Gaming the system may continue to work, if you go to work for a big company, depending on how broken the company is, you may be able to succeed by sucking up to the right person; Giving the impression of productivity by sending emails late at night, or if you're smart enough changing the clock on your computer, cause who's going to check the headers, right? I like an audience I can tell jokes to and they laugh. Over in the business school: "headers?" Okay, God this thing is being recorded, I just realized that.

Alright for now on we are sticking strictly to the script. But, in startups, that does not work. There is no boss to trick, how can you trick people, when there is nobody to trick? There are only users and all users care about is whether your software does what they want, right? They're like sharks, sharks are too stupid to fool, you can't wave a red flag and fool it, it's like meat or no meat. You have to have what people want and you only prosper to the extent that you do. The dangerous thing is, especially for you guys, the dangerous thing is that faking does work to some extent with investors.

If you’re really good at knowing what you’re talking about, you can fool investors, for one, maybe two rounds of funding, but it's not in your interest to do. I mean, you're all doing this for equity, you're puling a confidence trick on yourself. Wasting your own time, because the startup is doomed and all you’re doing is wasting your time writing it down. So, stop looking for the trick. There are tricks in startups, as there are in any domain, but they are an order of magnitude less important than solving the real problem. Someone who knows zero about fundraising, but has made something users really love, will have an easier time raising money than someone who knows every trick in the book, but has a flat usage graph.

Though, in a sense, it's bad news that gaming the system stops working now, in the sense that you're deprived of your most powerful weapons and, after all, you spent twenty years mastering them. I find it very exciting that there even exist parts of the world where gaming the system is not how you win. I would have been really excited in college if I explicitly realized that there are parts of the world where gaming the system matters less than others, and some where it hardly matters at all. But there are, and this is one of the most important thing to think about when planning your future. How do you win at each type of work, and what do you want to win by doing it?

That brings us to our fourth counterintuitive point, startups are all consuming. If you start a startup, it will take over your life to a degree that you cannot imagine and if it succeeds it will take over your life for a long time; for several years, at the very least, maybe a decade, maybe the rest of your working life. So there is a real opportunity cost here. It may seem to you that Larry Page has an enviable life, but there are parts of it that are defiantly unenviable. The way the world looks to him is that he started running as fast as he could, at age twenty-five, and he has not stopped to catch his breath since. Every day shit happens within the Google empire that only the emperor can deal with and he, as the emperor, has to deal with it. If he goes on vacation for even a week, a whole backlog of shit accumulates, and he has to bear this, uncomplaining, because: number one, as the company’s daddy, he cannot show fear or weakness; and number two, if you’re a billionaire, you get zero, actually less than zero sympathy, if you complain about having a difficult life.

Which has this strange side effect that the difficulty of being a successful startup founder is concealed from almost everyone who has done it. People who win the one-hundred meter in the Olympics, you walk up to them and they're out of breath. Larry Page is doing that too, but you never get to see it.

Y Combinator has now funded several companies that could be called big successes and in every single case the founder says the same thing, "It never gets any easier." The nature of the problems change, so you're maybe worrying about more glamorous problems like construction delays in your new London offices rather than the broken air conditioner in your studio apartment, but the total volume of worry never decreases. If anything, it increases.

Starting a successful startup is similar to having kids; it's like a button you press and it changes your life irrevocably. While it's honestly the best thing—having kids—if you take away one thing from this lecture, remember this: There are a lot of things that are easier to do before you have kids than after, many of which will make you a better parent when you do have kids. In rich countries, most people delay pushing the button for a while and I'm sure you are all intimately familiar with that procedure.

Yet when it comes to starting startups a lot of people seem to think they are supposed to start them in college. Are you crazy? What are the universities thinking – they go out of their way to ensure that their students are well supplied with contraceptives, and yet they are starting up entrepreneurship programs and startup incubators left and right.

To be fair, the universities have their hand forced here. A lot of incoming students are interested in start-ups. Universities are at least de-facto supposed to prepare you for your career, and so if you're interested in startups, it seems like universities are supposed to teach you about startups and if they don't maybe they lose applicants to universities that do claim to do that. So can universities teach you about startups? Well, if not, what are we doing here? Yes and no, as I've explained to you about start-ups. Essentially, if you want to learn French, universities can teach you linguistics. That is what this is. This is linguistics: we're teaching you how to learn languages and what you need to know is how a particular language.

What you need to know are the needs of your own users. You can't learn those until you actually start the company, which means that starting a startup is something you can intrinsically only learn by doing it. You can't do that in college for the reason I just explained. Startups take over your entire life. If you start a startup in college, if you start a startup as a student, you can't start a startup as a student because if you start a startup you're not a student anymore. You may be nominally a student but you won't even be that for very much longer. Given this dichotomy: which of the two paths should you take?

Be a real student and not start a startup or start a real startup and not be a student. Well, I can answer that one for you. I'm talking to my own kids here. Do not start a startup in college. I hope I'm not disappointing anyone seriously. Starting a startup could be a good component of a good life for a lot of ambitious people. This is just a part of a much bigger problem that you are trying to solve. How to have a good life, right. Those that are starting a startup could be a good thing to do at some point. Twenty is not the optimal time to do it.

There are things that you can do in your early twenties that you cannot do as well before or after. Like plunge deeply into projects on a whim that seem like they will have no pay off. Travel super cheaply with no sense of a deadline. In fact they are really isomorphic shapes in different domains.

For unambitious people your thing can be the dreaded failure to launch. For the ambitious ones it's a really valuable sort of exploration and if you start a startup at twenty and you are sufficiently successful you will never get to do it.

Mark Zuckerberg will never get to bum around a foreign country. If he goes to a foreign county, it's either as a de-facto state visit or like he's hiding out incognito at George V in Paris. He's never going to just like backpack around Thailand if that's still what people do. Do people still backpack around Thailand? That's the first real enthusiasm I've ever seen from this class. Should have given this talk in Thailand. He can do things you can't do, like charter jets to fly him to foreign countries. Really big jets. But success has taken a lot of the serendipity out of his life. Facebook is running him as much as he's running Facebook.

While it can be really cool to be in the grip of some project you consider your life's work, there are advantages to serendipity. Among other things, it gives you more options to choose your life's work from. There's not even a trade off here. You're not sacrificing anything if you forgo starting a start up at twenty because you will be more likely to succeed if you wait. In the astronomically unlikely case that you are twenty and you have some side project that takes off like Facebook did, then you face a choice to either be running with it or not and maybe it's reasonable to run with it. Usually the way that start ups take off is for the founders to make them take off. It's gratuitously stupid to do that at twenty.

Should you do it at any age? Starting a startup may sound kind of hard, if I haven't made that clear let me try again.

Starting a startup is really hard. If it’s too hard, what if you are not up to this challenge?

The answer is the fifth counter intuitive point. You can't tell. Your life so far has given you some idea of what your prospects might be if you wanted to become a mathematician or a professional football player. Boy, it’s not every audience you can say that to. Unless you have had a very strange life indeed you have not done much that’s like starting a startup. Meaning starting a startup will change you a lot if it works out. So what you’re trying to estimate is not just what you are, but what you could become. And who can do that? Well, not me. for the last nine years it was my job to try to guess (I wrote "predict" in here and it came out as "guess"—that’s a very informative Freudian slip). Seriously it’s easy to tell how smart people are in ten minutes. Hit a few tennis balls over the net, and do they hit them back at you or into the net? The hard part and the most important part was predicting how tough and ambitious they would become.

There may be no one at this point who has more experience than me in doing this. I can tell you how much an expert can know about that. The answer is not much. I learned from experience to keep completely open mind about which start ups in each batch would turn out to be the stars. The founders sometimes thought they knew. Some arrived feeling confident that they would ace Y Combinator just as they had aced every one of the few easy artificial tests they had faced in life so far. Others arrived wondering what mistake had caused them to be admitted and hoping that no one discover it.

There is little to no correlation between these attitudes and how things turn out. I've read the same is true in the military. The swaggering recruits are no more than likely to turn out to be really tough than the quiet ones and probably for the same reason. The tests are so different from tests in people’s previous lives. If you are absolutely terrified of starting a startup you probably shouldn’t do it. Unless you are one of those people who gets off on doing things you're afraid of. Otherwise if you are merely unsure of whether you are going to be able to do it, the only way to find out is to try, just not now.

So if you want to start a startup one day, what do you do now in college? There are only two things you need initially, an idea and cofounders. The MO for getting both of those is the same which leads to our sixth and last counterintuitive point.

The way to get start up ideas is not to try to think of startup ideas. I have written a whole essay on this and I am not going to repeat the whole thing here. But the short version is that if you make a conscious effort to try to think of startup ideas, you will think of ideas that are not only bad but bad and plausible sounding. Meaning you and everybody else will be fooled by them. You'll waste a lot of time before realizing they're no good. The way to come up with good startup ideas is to take a step back. Instead of trying to make a conscious effort to think of startup ideas, turn your brain into the type that has startup ideas unconsciously. In fact, so unconsciously that you don't even realize at first that they're startup ideas. This is not only possible: Yahoo, Google, Facebook, Apple all got started this way. None of these companies were supposed to be companies at first, they were all just side projects. The very best ideas almost always have to start as side projects because they're always such outliers that your conscious mind would reject them as ideas for companies.

How do you turn your mind into the kind that has startup ideas unconsciously? One, learn about a lot of things that matter. Two, work on problems that interest you. Three, with people you like and or respect. That's the third part incidentally, is how you get cofounders at the same time as the idea. The first time I wrote that paragraph, instead of learn a lot about things that matter, I wrote become good at some technology. But that prescription is too narrow.

What was special about Brain Chesky and Joe Gebbia from Airbnb was not that they were experts in technology. They went to art school, they were experts in design. Perhaps more importantly they were really good at organizing people in getting projects done. So you don't have to work on technology per se, so long as you work on things that stretch you.

What kinds of things are those? Now that is very hard to answer in the general case. History is full of examples of young people who were working on problems that no one else at the time thought were important. In particular that their parents didn't think were important. On the other hand, history is even fuller of examples of parents that thought their kids were wasting their time and who were right.

How do you know if you’re working on real stuff? I mean when Twitch TV switched from being Justin.tv to Twitch TV and they were going to broadcast people playing video games, I was like, "What?" But it turned out to be a good business. I know how I know real problems are interesting, and I am self-indulgent: I always like working on anything interesting things even if no one cares about them. I find it very hard to make myself work on boring things even if they're supposed to be important. My life is full of case after case where I worked on things just because I was interested and they turned out to be useful later in some worldly way.

Y Combinator itself is something I only did because it seemed interesting. I seem to have some internal compass that helps me out. This is for you not me and I don't know what you have in your heads. Maybe if I think more about it I can come up some heuristics for recognizing genuinely interesting ideas. For now all I can give you is the hopelessly question begging advice. Incidentally this is the actual meaning of the phrase begging the question. The hopelessly question begging advice that if you’re interested in generally interesting problems, gratifying your interest energetically is the best way to prepare yourself for a startup and probably best way to live.

Although I can't explain in the general case what counts as an interesting problem I can tell you about a large subset of them. If you think of technology as something that’s spreading like a sort of fractal stain, every point on the edge represents an interesting problem. Steam engine not so much maybe you never know. One guaranteed way to turn your mind into the type to start up ideas for them unconsciously. Is to get yourself to the leading edge of some technology. To, as Paul Buchheit put it, "Live in the future." And when you get there, ideas that seem uncannily prescient to other people will seem obvious to you. You may not realize they're start up ideas, but you will know they are something that ought to exist.

For example back at Harvard in the mid 90s. A fellow grad student of my friends Robert and Trevor wrote his own voice over IP software. It wasn't meant to be a startup, he never tried to turn it into one. He just wanted to talk to his girlfriend in Taiwan without paying for long distance calls. Since he was an expert on networks, it seemed obvious to him that thing to do was to turn the sound into packets and ship them over the internet for free. Why didn't everybody do this? They were not good at writing this type of software. He never did anything with this. He never tried to turn this into a startup. That is how the best startups tend to happen.

Strangely enough the optimal thing to do in college if you want to be a successful startup founder is not some sort of new vocational version of college focused on entrepreneurship. It's the classic version of college is education its own sake. If you want to start your own startup what you should do in college is learn powerful things and if you have genuine intellectual curiosity that’s what you’ll naturally tend to do if you just follow your own inclinations. The component of entrepreneurship, can never quite say that word with a straight face, that really matters is domain expertise. Larry Page is Larry Page because he was an expert on search and the way he became an expert on search was because he was genuinely interested and not because of some ulterior motive. At its best starting a startup is merely a ulterior motive for curiosity and you’ll do it best if you introduce the ulterior motive at the end of the process. So here is ultimate advice for young would be startup founders reduced to two words: just learn.

Alright how much time do we have left? Eighteen minutes for questions good god. Do you guys have the questions?

Q: Sure we will start with two questions. How can a nontechnical founder most efficiently contribute to a startup?

A: If the startup is, if the startup is working in some domain, if it’s not a pure technology startup but is working in some very specific domain, like if it is Uber and the non technical founder was an expert in the limo business then actually then the non technical founder would be doing most of the work. Recruiting drivers and doing whatever else Uber has to do and the technical founder would be just writing the iPhone app which probably less, well iPhone and android app, which is less than half of it. If it’s purely a technical start up the non technical founder does sales and brings coffee and cheeseburgers to the programmer.

Q: Do you see any value in business school for people who want to pursue entrepreneurship?

A: Basically no, it sounds undiplomatic, but business school was designed to teach people management. Management is a problem that you only have in a startup if you are sufficiently successful. So really what you need to know early on to make a start up successful is developing products. You would be better off going to design school if you would want to go to some sort of school. Although frankly the way to learn how to do it is just to do it. One of the things I got wrong early on is that I advised people who were interested in starting a startup to go work for some other company for a few years before starting their own. Honestly the best way to learn on how to start a startup is just to just try to start it.

You may not be successful but you will learn faster if you just do it. Business schools are trying really hard to do this. They were designed to train the officer core of large companies, which is what business seemed to be back when it was a choice to be either the officer core of large companies or Joe's Shoe Store. Then there was this new thing, Apple, that started as small as Joe's Shoe Store and turns into this giant mega company but they were not designed for that world they are good at what they're good at. They should just do that and screw this whole entrepreneurship thing.

Q: Management is a problem only if you are successful. What about those first two or three people?

A: Ideally you are successful before you even hire two or three people. Ideally you don't even have two or three people for quite awhile. When you do the first hires in a startup they are almost like founders. They should be motivated by the same things, they can't be people you have to manage. This is not like the office, these have to be your peers, you shouldn't have to manage them much.

Q: So is it just a big no no, someone has to be managed no way they should be on the founding team.

A: In the case were you are doing something were you need some super advanced technical thing and there is some boffin that knows this thing and no one else in this world including on how to wipe his mouth. It may be to your advantage to hire said boffin and wipe his mouth for him. As a general rule you want people who are self motivated early on they should just be like founders.

Q: Do you think we are currently in a bubble?

A: I'll give you two answers to this question. One, ask me questions that are useful to this audience because these people are here to learn how to start startups, and I have more data in my head than anybody else and you're asking me questions a reporter does because they cannot think of anything interesting to ask. I will answer your question. There is a difference between prices merely being high and a bubble. A bubble is a very specific form of prices being high where people knowingly pay high prices for something in the hope that they will be able to unload it later on some greater fool. That's what happened in the late 90's, when VC's knowingly invested in bullshit startups thinking that they would be able to take those things public and unload them on other retail investors before everything blew up

I was there for that at the epicenter of it all. That is not what is happening today. Prices are high, valuations are high, but valuations being high does not mean a bubble. Every commodity has prices that go up and down in some sort of sine wave. Definitely prices are high. We tell people if you raise money, don't think the next time you raise money it's going to be so easy, who knows maybe between now and then the Chinese economy will have exploded then there's a giant disaster recession. Assume the worst. But bubble? No.

Q: I am seeing a trend among young people and successful entrepreneurs where they don't want to start one great company but twenty. You are starting to see a rise in these labs attempts were they are going to try to launch a whole bunch of stuff, I don't have any stellar examples yet.

A: Do you mean like IDEO?

Q: No, like Idealab, Garrett Camp's new one...

A: Oh yeah. There's this new thing were people start labs that are supposed to spin off startups. It might work, that's how Twitter started. In fact, I meant Idealab, not IDEO, that was another Freudian slip. Twitter was not Twitter at first. Twitter was a side project at a company called Odeo that was supposed to be in the podcasting business, and you like podcasting business, do those words even grammatically go together? The answer turned out to be no as Evan discovered. As a side project they spun off Twitter and boy was that a dog wagging tail, people are starting these things that are supposed to spin off startups, will it work? Quite possibly if the right people do it. You can't do it though, because you have to do it with your own money.

Q: What advice do you have for female co-founders as they are pursuing funding?

A: It probably is true that women have a harder time raising money. I have noticed this empirically and Jessica is just about to publish a bunch of interviews on female founders and a lot of them said that they thought they had a harder time raising money, too. Remember I said the way to raise money? Make your start up actually do well and that's just especially true in any case if you miss the ideal target from the VC's point of view in any respect. The way to solve that problem is make the startup do really well. In fact, there was a point a year or two ago when I tweeted this growth graph of this company and I didn't say who they were. I knew it would get people to start asking and it was actually a female founded startup that was having trouble raising money, but their growth graph was stupendous. So I tweeted it, knowing all these VC's would start asking me, “Who is that?” Growth graphs have no gender, so if they see the growth graph first, let them fall in love with that. Do well, which is generally good advice for all startups.

Q: What would you learn in college right now?

A: Literary theory, no just kidding. Honestly, I think I might try and study physics that’s the thing I feel I missed. For some reason, when I was a kid computers were the thing, maybe they still are. I got very excited learning to write code and you can write real programs in your bedroom. You can't build real accelerators, well maybe you can. Maybe physics, I noticed I sort of look longingly at physics so maybe. I don't know if that’s going to be helpful starting a startup and I just told you to follow your own curiosity so who cares if it's helpful, it'll turn out to be helpful.

Q: What are your reoccurring systems in your work and personal life that make you efficient?

A: Having kids is a good way to be efficient. Because you have no time left so if you want to get anything done, the amount of done you do per time is high. Actually many parents, start up founders who have kids have made that point explicitly. They cause you to focus because you have no choice.

I wouldn't actually recommend having kids just to make you more focused. You know, I don't think I am very efficient, I have two ways of getting work done. One is during Y Combinator, the way I worked on Y Combinator is I was forced to. I had to set the application deadline, and then people would apply, and then there were all these applications that I had to respond to by a certain time. So I had to read them and I knew if I read them badly, we would get bad startups so I tried really hard to read them well. So I set up this situation that forced me to work. The other kind of work I do is writing essays. And I do that voluntarily, I am walking down the street and the essay starts writing itself in my head. I either force myself to work on less exciting things; I can't help working on exciting things. I don't have any useful techniques for making myself efficient. If you work on things you like, you don't have to force yourself to be efficient.

Q: When is a good time to turn a side project into a startup?

A: You will know, right. So the question is when you turn a side project into a startup, you will know that it is becoming a real startup when it takes over a alarming large percentage of your life, right. My god I've just spent all day working on this thing that’s supposed to be a side project, I am going to fail all of my classes what am I going to do, right. Then maybe it’s turning into a startup.

Q: I know you talked a lot, earlier, about you'll know when your start up is doing extremely well, but I feel like in a lot of cases it's a gray line, where you have some users but not explosive growth that is up and to the right, what would you do or what would you recommend in those situations? Considering allocating time and resources, how do you balance?

A: When a start up is growing but not much. Didn't you tell them they were supposed to read Do Things that Don't Scale? You sir have not done the readings, you are busted. Because there are four, I wrote a whole essay answered that question and that is to do things that don't scale. Just go read that, because I can't remember everything I said. It's about exactly that problem.

Q: What kind of startup should not go through incubation, in your opinion?

A: Definitely any that will fail. Or if you'll succeed but you're an intolerable person. That also Sam would probably sooner do without. Short of that, I cannot think of any, because a large percentage, founders are often surprised by how large a percentage of the problems that start ups have are the same regardless of what type of thing they're working on. And those tend to be kind of problems that YC helps the most not the ones that are domain specific. Can you think of the class of startups? That YC wouldn't work for? We had fission and fusion startups in the last batch.

Q: You mentioned that it's good advice to learn a lot about something that matters, what are some good strategies to figure out what matters?

A: If you think of technology as something that’s spreading as a sort of fractal stain. Anything on the edge represents an interesting idea, sounds familiar. Like I said that was the problem, you have correctly identified the thing I didn't really answer the question were I gave this question begging answer. I said I'm interested in interesting things and you said you were interested in interesting things, work on them and things will work out.

How do you tell what is a real problem? I don't know, that's like important enough to write a whole essay about. I don't know the answer and I probably should write something about that, but I don't know. I figured out a technique for detecting whether you have a taste for generally interesting problems. Which is whether you find working on boring things intolerable and there are known boring things. Like literary theory and working in middle management in some large company. So if you can tolerate those things, then you must have stupendous self-discipline or you don't have a taste for genially interesting problems and vice versa.

Q: Do you like Snapchat?

A: Snapchat? What do I know about Snapchat? We didn't fund them. I want another question.

Q: If you hire people you like, you might get a monoculture and how do you deal with the blind spots that arise?

A: Starting a startup is where many things will be going wrong. You can't expect it to be perfect. The advantage is of hiring people you know and like are far greater than the small disadvantage of having some monoculture. You look at it empirically, at all the most successful startups, someone just hires all their pals out of college.

Alright you guys thank you.

Lecture 4: Building Product, Talking to Users, and Growing

Adora Cheung

Thanks for having me. Today I am going to be talking about how to go from zero users to many users. I'm just assuming that you have many great ideas in your head at this moment and you are thinking about what the next step is.

A lot of my lecture is based off of mistakes I have made in the past. As Sam mentioned, I went to YC in 2010 and spent three years going back and forth, pivoting a bunch of times, starting over a bunch of times, and I learned a lot about what not to do if I were to start another startup after Homejoy. A lot of my advice comes from failure and understanding what you shouldn't do and then using that to make generalizations about what you should do.

Just a reminder that you should take all advice as directionally good guidance, but every business is different. You're different, and I'm not you, so take everything with that in mind.

When you start a startup you should have a lot of time on your hands to concentrate on the startup. I'm not saying that you should quit school or quit work; what I'm saying is that you should have a lot of compressed time that is dedicated to immersing yourself in the idea and developing solutions to the problem that you are trying to solve. For example, if you're in school it is better to have one or two days straight per week to work on your idea versus spending two hours here and there every single day during the course of the week. It's like coding. There is a lot of context switching so being able to really focus and immerse yourself is really important.

When I first wrote this lecture I was thinking, what are the things that most people do incorrectly when starting a startup? The novice approach is thinking, "I have this really great idea, I don't want to tell anyone about it. I'm going to build, build, build and then going to maybe tell one or two people and then I'm going to launch it on TechCrunch or somewhere like that, and then I'm going to get lots of users."

What really happens is because you did not get a lot of feedback, maybe you get a lot of people to your site, but no one sticks around because you didn't get that initial user feedback. If you're lucky enough to have some money in the bank you might go buy some users but it just whittles out over time and you just give up. It is sort of a vicious cycle. I actually did this once, and I did this while I was in YC. When I went through YC I didn't even launch a product. I didn't launch on TechCrunch which is the thing you should definitely do. You don't ever want to get into that cycle because you'll just end up with nothing good.

The next thing is that you have an idea and you should really think about what the idea is really solving. Like what is the actual problem. You should be able to describe your problem in one sentence. And then you should think, "How does that problem relate to me? Am I really passionate about that problem?" And then you should think, "Okay it's a problem I have, but is it a problem that other people have?" And you verify that by going out and talking to people.

One of the biggest mistakes I've made involves my co-founder and I, who is also my brother. We started a company called Pathjoy in 2009 or 2010. We had two goals in mind. One was to create a company that made people really happy, and to create a company that was very, very impactful. A good proxy for that is to just create a big huge company. And so we thought, okay, the problem we are solving is to make people happier. We first went to the notion of who are the people who make people happy. We came up with life coaches and therapists. It seemed kind of obvious to create a platform for life coaches and therapist. What happened as a result was that when we started using the product ourselves, we aren't cynical people by any means, but life coaches and therapists are just not people we would use ourselves. It was sort of useless to us. So it wasn't even a problem that we had and it wasn't something that we were super passionate about building out, yet we spent almost a year trying to do this. And so if you just start from T=0 and think about this before you build any product I think you can save yourself a lot of headache down the road from doing something you don't want to do.

So say you have a problem and you are able to state it, where do you start and how do you think of solutions? The first thing you should do is think about the industry that you are getting yourself into. Whether it is big or whether it is huge, you should really immerse yourself in that industry. And there are a number of ways to do this.

One is to really become a cog in that industry for a little bit. And so it might seem a little counterintuitive to do this because most people say that if you really want to disrupt an industry you should really not be a player in it. Someone who spent 20 or 30 years in an industry is probably set in their ways and is just used to the way things work and really can't think about what the inefficiencies are or the things that you can "disrupt". However, as a newbie coming into the industry you really should take one or two months to just really understand what all of the little bits and pieces of the industry are and how it works. Because it's when you get into the details, that's when you start seeing things that you can be exploiting and things that are really inefficient and may provide a huge overhead cost that you may be able to cut down.

So an example of this is that when we started Homejoy, we started with the cleaning industry, and when we started we were the cleaners ourselves. We started to clean houses and we found out really quickly that we were very bad cleaners. As a result, we said okay, we have to learn more about this and we went to buy books. We bought books about how to clean, which helped maybe a little bit. We learned a little more about cleaning supplies but it is sort of like basketball, you can read and learn about basketball but you're not going to get better at it if you don't actually train and throw a basketball into the net.

And so we decided that one of us was going to have to learn how to clean. Or at least get trained by a professional. We actually went to get a job at a cleaning company itself. The cool thing was I learned how to clean from training the few weeks that I was there at the cleaning company, but the even better thing was that I learned a lot about how a local cleaning company works. In that sense I learned why a local cleaning company could not become huge like Homejoy is today. And that is because they are pretty old school and they have a lot of things that are done inefficiently. Such as booking the customer and optimizing the cleaners' schedules was just done very inefficiently.

If you are in a situation like mine where there is a service element of it then you should go and do that service yourself. If your thing is related to restaurants you should become a waiter, if it is related to painting become a painter and kind of get in the shoes of your customers from all angles of what you are trying to build.

The other thing is there is also a level of obsessiveness that you should have with it as well. You should be so obsessed that you want to know what everybody in that space is doing. And it is things like writing a list of all of the potential competitors, similar types of companies, and Google searching them and clicking on every single link and reading every single article from search result number 1 to 1000. I found all potential competitors big and small and if they were public, I would go and read their S-1s, I would go read all of their quarterly financials, I would sit on earnings calls. You know most of these, you don't get much out of it but there are these golden nuggets that you will find every once in a while. And you won't be able to find that unless you actually go through the work of getting all that information in your head. You should become an expert in your industry. There should be no doubt when you are building this that you are the expert so that people trust you when you are building this product.

The second thing is identifying customer segments. Ideally at the end of the day you have built a product or business that everybody in the world is using. In the beginning, you realistically want to corner off a certain part of the customer base so that you can really optimize for them. It is just about focus and whether you are catering to teenage girls or whether it is soccer moms, you will be able to focus a lot on their needs.

And lastly, before you even create a product or before you put code down, you should really storyboard out the user experience of how you are going to solve the problem. And that is not just meaning the website itself, it also means how does the customer find out about you. It can be through an ad or word-of-mouth, and then they come to your site and they learn more about you. What does that text say and what are you communicating to them when they sign up for the project and when they purchase the service? What are they actually getting from your service or product? After they finish using the product or service do they leave a review or do they leave comments? You need to be able to go through that whole flow and visualize in your head what the perfect user experience is. And then put it down on paper and put it into code, and then start from there.

So, you have all these ideas in your head, now you kind of know what the core customer base is that you want to go after, and you know everything about the industry, what do you do next? You start building your product. The common phrase that most people use today is," You should build a minimum viable product." And I underlined viable because I think a lot of people skip that part and they go out with a feature and the whole user experience in the very beginning is flat. Minimal viable product pretty much means what is the smallest feature set that you should build to solve the problem that you are trying to solve. I think if you go through the whole storyboarding experience you can kind of figure that out very quickly. But again, you have to be talking to users, you have to be seeing what exists out there already, and what you should be building should solve their immediate needs.

And the second thing is that before you put things in front of the user you should really have your product positioning down. What I mean by that is that you should be able to go to a person and be able to say, "Hey, this does X,Y, and Z in one sentence." So for example, at Homejoy we started off with something super complicated. We were an online platform for home services, you start with cleaning and you can choose blah blah blah. It just went on for paragraphs and paragraphs.

When we went to potential users to come on our platform they would kind of get bored after the first few sentences. What we found out was that we needed a one-liner. The one-liner was very important. It kind of describes the functional benefits of what you do. In the future when you are trying to build a brand or whatnot you should be able to describe the emotional benefits and stuff like that. But when you are starting with no users you really need to tell them what they are going to get out of it. After we changed our position to get your place cleaned for $20 an hour, then everyone got it and we were able to get users in the door that way.

So you have an MVP out there, now how do you get your first few users to start trying it? The first few users should be obviously people you are connected with. You and your cofounder should be using it, your mom and dad should be using it, and your friends and coworkers should be using it. Beyond that, you want to get more user feedback. I've listed here some of the obvious places to go to depending on what you are selling. You can take your pick of the draw here. So, online communities, on Hacker News now there is the show HN - that's a great place. Especially if you are building tools for developers and things like that. Local communities - so if you're building consumer products you know there are a lot of influential local community mailing lists. Especially those for parents. Those are places you might want to hit up too.

At Homejoy we actually tried all of these. We used it ourselves and that was fine. We were the only cleaners so that was pretty easy. Our parents live in Milwaukee and we were based in Mountain View so that didn't work. Friends and coworkers were kind of like in San Francisco and elsewhere so we didn't have too many of them use it. So we actually ended up in a dead end of not being able to convince many people to use it in the beginning. So what we did was, because we are in Mountain View, some of you guys might know on Castro Street they have street fairs there during the summertime. So we would go out and basically chase down people and get them to try to book a cleaning. Almost everyone would say no until one day we just took advantage of the weather. It was a very hot and humid day and what we noticed was that everyone gravitated towards the food and drink area, especially on a hot day.

We figured we needed to get in the middle of that so we took water bottles and froze them and we started handing out free bottles of water that were cold. And people just came to us. I think we basically guilt tripped people into booking cleanings. But the proof in the pudding was that I figured most of the people were guilt tripped into doing it, but then they went home and they didn't cancel on us. Well, some of them did but the majority of them did not. I thought that's good, I have to go clean their houses but at least there is something we are actually solving here.

I know another startup in the last batch, I forgot their name right now, but they were selling shipping type products or trying to replace shipping products. So they would show up to the US postal office and find people who were trying to ship products and just take them out of line and get them to try to use the product and have them ship it for them. So you just have to go to places where people are really going to show up. Your conversion rate is going to be really low but to go from 0 to 1 to 3 to 4 these are the kind of things you might have to do.

So now that you have users using you ,what do you do with all of these users? The first thing you should do is make sure that there is a way for people to contact you. Ideally there is a phone number and if you put up a phone number, one good idea is to make sure that you have a voicemail so that you won't be picking it up all the time. But in any case a way for people to give inbound feedback is good, but really what you should be doing is going out to your users and talking to them. Get away from your desk and just get out and do the work. It seems like a slog and it is going to be a slog but this is where you are going to get the best feedback ever for your product. And this is where it is going to teach you what features you need to completely change, get rid of, or what features you need to build.

One way to do this is to send out surveys to get reviews after they have used the product. This is okay but generally people are only going to respond if they really love you or they really hate you. And you never get the in between. A way to get the in between and not all of the extremes is to actually meet the person that is using your product. I've seen people go out to meet the user and they sit there and it is like a laboratory and it is like an inquisition. You're just kind of poking at them. That is not going to give you the best results. What you should really do is make it into a conversation and get to know them and get them to feel comfortable. You want to get them at a level where they feel like they should be honest with you to help you improve things. So I found that actually taking people out for drinks and stuff like that was actually a very good way to do that. I'm not sure if all of you are old enough to do that but you can take them for coffee.

So another thing that you should be tracking is how are you doing in general from a macro perspective. The best way to do that is by tracking customer retention. The number of people that came in the door today, the number of people who are coming back tomorrow, the next day and so forth. Usually over time you are kind of looking at monthly retention so people who came in the door today, are they still using it next month and so forth. The problem with that metric is that it takes forever to collect that data and sometimes you don't have a month or two months or three months to figure that out. So a good leading indicator is actually collecting reviews and ratings. Such as five-star and four-star reviews or collecting some notion of nps, which is net promoter score. So you're basically asking them for a rating from 0 to 10 about how likely are they to recommend you to a friend and calculating the nps.

Over time what you'll see is that as you are building new features, you will be able to see that the reviews and the retention are going up over time. That means that you are doing a good job. If it is going down then you are doing a bad job. If it is kind of staying the same that probably means that you need to go out and figure out what new things you should be building.

One thing you should be wary of is the honesty curve. Some people will just lie to you. These are degrees of separation from you, and this is the level of honesty. So here this is your mom, these are the friends of your friends and here are random people. Your mom will use your product and she will be proud of you anyway, so she'll be honest this much. Your friends will be pretty honest with you and give you feedback because they care about you - this is assuming this is a free product - and then over time as you get more and more random, these people don't know who you are. There are people over here who don't care about giving you feedback. So take this into consideration when getting user feedback.

So say now this is a paid product. So when it is a paid product your mom is down here. She is just going to lie to you and tell you it's great. But then it kind of goes like this (draws graph going upward). Your friends are going to support you and give you the right feedback but it is actually these random people out here that if they really don't think that what they paid for was worth it, they are going to really tell you. That's because it is money out the door.

This is another way of saying that you are going to get the best feedback if you just make someone pay for it. That's not to say that you should make people pay for it the first time out, but it is to say that if you are going to build a product that you are going to eventually need to pay for the software or for the hardware or whatever then get to the point where you can do that very fast. Because that is when you can get to the more meaty stuff of how you can get more paying users in the door.

You're getting a lot of feedback and what do you do before you officially launch the product? You always want to be building fast and you want to be optimizing for this stage of your growth. You might have 10 users at this point and there is no point in trying to build features for when you might have 10 million users. You want to optimize for the next stage of growth which will be 10 to 100 users. What are the features you really need for that and just go with that. One of the things I found when building a marketplace is that process is very important over time as you scale.

You need to not try and automate everything and create software to have robots run everything. What you should do to really understand what you should build is manually do it yourself. An example of this is when we started taking on cleaning professionals on to our platform, we would ask them a bunch of questions over the phone and then in person would ask a bunch of questions as well. And then they would go to a test clean and then they would get onboarded to our platform if they were good enough. Doing all these questions for that many candidates we had a 3-5% acceptance rate.

What happened over time was that we learned certain questions that we were asking were good indicators as to whether or not they would be a good or bad performer on the platform through data collection and just looking at everything we could ask on an online form. That is when we put up an online application, they could apply and then we would ask them maybe several other questions during the in person interview. If you try to automate things too fast then you run into this potential problem of not being able to move quickly and iterate things like questions on an application and things like that.

A third point here is temporary brokenness is much better than permanent paralysis. By that what I mean is perfection is irrelevant during this stage. When you get to the next stage of growth what you are trying to perfect in one stage is not going to matter anyway. So do not worry about all of the edge cases when you are building something, just worry about the generic case of who your core user is going to be. As you get bigger and bigger the volume of those edge cases increases over time and you will want to build for that.

Lastly beware of the Frankenstein approach which is - great you talked to all of these users and they gave you all of these ideas and the first thing you are going to want to do is go build every single one of them and then go show them the next day and make them happier. You should definitely listen to user feedback but when someone tells you to build a feature you shouldn't go build it right away. What you should really do is get to the bottom of why they are asking you to build the feature. Usually what they are suggesting is not the best idea. What they are really suggesting is that I have this other problem that you either created for me while using the product or I really need this problem solved if I'm going to pay to use this product. So figure that out first before piling on a bunch of features which then hide the problem altogether.

So you have a product that you are ready to ship - some people at this point will continue building the product and not ship it at all. I think the whole idea of being stealth and perfecting the product to no end is the idea that imitation is cheaper than innovation in terms of time and money and capital. I think that everyone should always assume in general that if you have a really good idea no matter when you launch someone is going to fast follow you and someone is going to execute as hard as they possibly can to catch up with you. There is no point in holding out on all of that user feedback that you can get by getting a lot of users because he felt paranoid that someone is going to do this to you.

I hate to keep harping on it but these are things that I see today with founders and something that I went through as well. And I think that unless you are building something that requires tens of millions of dollars just to start up there is really no point in waiting around to launch the product.

So say you have something that you feel ready to get lots of users on. So what do you do at this point? I will go over various types of growth in the next slides, but the one thing to note here early on when it is just you, your cofounder, and a couple of other people building, you aren't creating a team just for growth. It is going to be one person and one person only. You really need to focus and you are going to be tempted to try five different strategies at one time.

But really what you should do is take one channel and really execute on it for an entire week and just focus on that. And if that works continue executing on it until it caps out. If it doesn't work then just move on. By doing this you will feel more certain that the channel you were working on is wrong and your initial hypothesis is wrong than if you only spent a third of your time on it over the course of a few weeks. So learn one channel at a time.

Second, when you find one channel at a time and strategies that work, always be iterating on it. You can potentially create a playbook and give it to someone else to iterate on it but these channels always change. Anything from Facebook ads to Google ads, the distribution channels, the environments that you don't control change all of the time and you should always be iterating and optimizing for that. And lastly, in the beginning when you see a channel that fails just to get rid of it and go on there are lots of other things to try. But over time go back to that channel and look at it again.

An example is that in the beginning at Homejoy we had no money so when we tried to buy Google ads to get users in the door quickly - what we found was that all of these national companies had more money than us, they were making a lot more money on the job than us. So they were able to acquire users at a much higher cost than us. So we couldn't afford that and we had to go through another channel. But today we make more money on the job, and we are better at some things. So we should probably revisit the idea of buying Google ads. That's what I mean by that.

And the key to all of this is creativity. Performance marketing, or marketing and growth in general can be very technical but, it is actually technical, and you have to be creative because if it was really easy and bland then everyone would be growing right now. So you always have to find that little thing that no one else is doing and do that to the extreme.

So there are three types of growth. Sticky, viral, and paid growth. Sticky growth is trying to get your existing users to come back and pay you more or use you more. Viral growth is when people talk about you. So you use a product, you really like it and you tell ten other friends, and they like it. That's viral growth. And the third is paid growth. If you happen to have money in the bank you're going to be able to use part of that money to buy growth.

The central theme that I'm going to go through is sustainability. By sustainable growth I mean you are basically not a leaky bucket. The money you put in has a good return investment on it. So sticky growth is, like I said, trying to get your existing users to come back and buy stuff. The only thing that really matters here is that you deliver a good experience. Right? If you deliver a good experience people are going to want to keep using you. If you deliver an addictive experience people are going to want to keep using you. And the way to measure this and to really look at this and how you are doing over time with whether you are providing good sticky growth is to look at the CLV and retention cohort analysis.

CLV, some people call it TLV, is a customer's lifetime. It is basically the net revenue that a customer brings in the door over a period of time. So a 12 month CLV is how much net revenue does a customer give you over 12 months. And sometimes people will do the month and six months and so forth. So when I say cohort basically what you are looking at is, this is time, and this is percent of the users coming back to you. So at period zero you are at 100%.

So cohort is another name for customer segments. For example you might look at the female versus male cohorts or people in Atlanta, Georgia versus people in Sacramento, California cohorts. The most common one is by month. So cohort equals month and let's just say for this exercise we are looking at March 2012. So in March 2012, 100% of the people are using your product. Now, one month later 50% of the people might come back. Now, in the second month how many people that came in March are coming back two months later? That might be down. So over time you will have a curve that looks like this. There is always some initial drop off. The reasons that people don't stay after first use could be that it wasn't worth it or they had a bad experience, or something like that. And then over time what you want is for your curve to flatten out. These over here become your core customers. These are the ones that will stay with you over time.

Say we are at one year later and you have built a bunch of stuff. You graph out the same thing and hopefully what you see is that you have a curve like this. That is, that even in the first period more than 50% of the people came back to you and more and more people are sticking with you. A really bad retention curve looks like this - which is after the first use they just hate you so much that no one even comes back. I don't know what kind of business that is, it is obviously a shitty business. I can't explain a good business that has a retention curve like that. Over time as you are thinking of strategies to increase this curve and to keep making it go up and up and up you want to keep looking at this analysis over time to see if that strategy is working for you.

The second kind of growth is viral growth. Like sticky growth you also need to deliver a good experience. But on top of that you need to deliver a really, really good experience. What is going to make these people shout out loud on Twitter or on Facebook or whatever and tell all their friends and email all of their family about you. You have to really deliver a good experience. Combined with that is you have to have really good mechanics for the referral program itself. You have 100 customers who really want to talk about you. Now how are they going to talk about you?

So in that sense the viral growth strategy is all about building a good experience, but if you have that, how do you build a good referral program. I have listed the three main parts of that. One is the customer touch points which is where are people learning that they can refer other people? That might be after they book or after they sign up. A better one is after they use the product for a while and you see that they are highly engaged, then you show them that link and get them to send it out to everyone. Another one is if you are doing more of a platform type play - for Homejoy we actually go inside their home. So another customer touch point is when the cleaning professional is inside the home they can have a leave behind and we can show them something there too as well. You want to basically put the customer touch points and the actual link to however they are going to refer their friends at a point in time where they are highly engaged and you know that they are loving you.

The second is program mechanics. The most common thing I have seen is $10 for $10. You get $10 if you invite your friends and they use it and they get $10. And so you should try different types of mechanics in that sense and try to optimize for whatever works for you. It could be 25 for 25 or it could be 10 for 10, it could be any of these things. And lastly, when your friend clicks on your referral link, when they come back to the site it is really important to optimize that conversion flow of how they are going to sign up. Sometimes you need to sell them in a different manner or up-sell that a friend suggested that you use this and so forth. So with all of these combined, you will really need to play around with them in different dimensions and come up with a good referral program.

And lastly is paid growth. Some examples of paid growth are this right here. And these are some of the most obvious ones and I'm sure that you guys can think of more. Paid growth is you happen to have money you can spend - you may have credit cards or whatever - but you can spend something to get users. So the correct way to think about paid growth is that you are going to risk putting money out there so that are you going to get a return. The simple way to think about it - is your CLV, your customer's lifetime - is it more than your CAC. And your CAC is an abbreviation for customer acquisition costs. So an example is - say you run a bunch of ads over 12 months and the customer is worth $300 to you. Each one of these ads, when you click on it the CPC costs different types of money, and then when they click on your ad they have to come to the site and sign up or buy something.

And the conversion rates are different for all of these ads. The CAC is calculated by the CPC divided by the conversion. So you see that there are different acquisition costs for different types of ads. To determine whether or not that is a good or bad ad all you have to do is CLV minus the CAC. If it is more than zero you are earning a profit. So you see that despite the CLV remaining the same and the conversions being higher or lower sometimes some ads that might seem good actually don't seem so good at the end of the day.

You can look at this for your whole entire customer base, aggregating all of your customers together, but the better way of looking at it is to break it down by customer segments. If you are building a marketplace for country music the CLVs of someone in Nashville, Tennessee is going to be much larger than the CLVs of someone in Czechoslovakia. I just assume that is the case anyway.

You will want to make sure that when you are buying ads for these different types of cohorts that you know what the differences are and you don't want to mix everything together. The last point on payback and sustainability - I think a lot of businesses get in trouble and they turn into bad businesses when they start spending beyond their means. And it has a lot to do with risk tolerance or how much risk you are willing to take on.

So when you look at these CLVs, which is suppose you get a customer that is worth $300 after 12 months. In the first month they are worth $100. If you wait until the 12 month period then they give you the other $200. But if in the first period you are actually paying $200 for them then you are in the hole for $100 until the end of the 12 month period. That's when you start to get into potentially unsustainable growth. Something could happen at the end of the 12 months where you don't actually get the $200 from the customer and you end up in a very bad situation. Essentially, at the end of the day you could be running out of money. And if you are doing this with credit cards you will definitely find that you are going to have to declare bankruptcy very soon.

So again, payback time is very important. Safe time to go with is three months. If you are very risk loving then maybe 12 months is better. Beyond 12 months is very much unsafe territory.

The art of pivoting - Homejoy in its current concept was literally the 13th idea we fully built out and tried to execute on and tried to get customers for. And so a lot of the questions I get are," How do you even get to that 13th idea, and how did you decide when to move on?" The best guidance that I can give on that is the kind of look at these three criteria, which is once you realize that you can't grow, and despite building out all of these great features and talking to all of these users none of them stick, or the economics of the business just don't make sense - then once you make that realization you just need to move on.

I think the trickiest one is probably the growth one because there are so many stories out there where the founders stuck with the idea and then after three years all of a sudden it started growing. So the trick here is what you really should do is have a growth plan when you start out. What is an optimistic but realistic way to grow this business? it might look something like this. In week one you just want one user, in week two you want maybe two users and so forth. And you can keep doubling up and up.

In week one you should basically build as much as possible to get that one user. And then a week to build as much to get two users. If you have a product that people want you should be able to maintain this growth curve pretty easily by just walking around and manually finding people. It is when you get to 100 users a week when you need these growth strategies to start working. What I tell people is usually if you are fully executing on your product, and you are working really hard, then if you go three or four weeks in a row of no growth or backwards growth, then it is time to maybe consider a pivot.

Maybe not in the sense that you completely come up with a new idea but you are probably fundamentally doing something wrong because at that early stage a startup should always be growing. This is optimistically what it looks like and this is the kind of growth curve that I set forth and put out when I started Homejoy, but really what it looks like is like this. So you want to make sure that when you are in a lull you don't stop. And that is what you should wait 2 to 3 weeks. As long as you don't stop working hard you'll eventually get back here and you'll see a trend like this over time.

I can take questions at this time.

Q: So one question online was if your users already have a product that they are already comfortable with how do you get them to switch to yours?

A: There is always a switchover cost. I will tell you the example of Homejoy. We were actually creating a new market in the sense that a lot of our initial users had never had cleanings before so it was pretty simple to get them on board. And a lot of people who have cleaners already really trust their cleaner. To get them to come and use something else is probably the most difficult task in the world. When you are building things and trying to get people to switch over to you what you really need to do is find the moments where your product or what you are offering is much better or very much differentiated from the existing solution they have.

So an example is someone who had a regular cleaner and maybe had a party one day and they needed a cleaning almost the next day. Because Homejoy in most areas has next-day availability they would just come to Homejoy and use it because they knew they couldn't get their regular cleaner. And once they start using the product, then that is when they start realizing the little advantages of using Homejoy, which adds up to a big advantage. Realizing that leaving cash out or using checks was really annoying so being able to do all of your payments online was more convenient. Being able to cancel or reschedule according to your own schedule was very convenient.

A lot of people when they build a product they are like - and these 50 things are better than the existing solution - and even if the benefits outweigh the switchover cost it is really hard to actually tell that to a user and try to get them to aggregate all of those benefits over many little things. It is better to have one or two things that clearly differentiate yourself from the other product.

Lecture 5: Business Strategy and Monopoly Theory

Peter Thiel

Sam: Alright, good afternoon, today's speaker is Peter Thiel, Peter was the founder of PayPal, Palantir, and Founders Fund and has invested in most of the tech companies in Silicon Valley. He's going to talk about strategy and competition. Thank you for coming, Peter.

Peter: Awesome, thanks Sam for inviting me, thanks for having me.

I sort of have a single idée fixe that I'm completely obsessed with on the business side which is that if you're starting a company, if you're the founder, entrepreneur, starting a company you always want to aim for monopoly and you want to always avoid competition. And so hence competition is for losers, something we'll be talking about today.

I'd like to start by saying something about the basic idea of when you start one of these companies, how do you go about creating value? What makes a business valuable? And I want to suggest there's basically a very simple formula, that if you have a valuable company two things are true. Number one, that it creates "X" dollars of value for the world. Number two, that you capture "Y" percent of "X." And the critical thing that I think people always miss in this sort of analysis is that "X" and "Y" are completely independent variables, and so "X" can be very big and "Y" can be very small. "X" can be an intermediate size and if "Y" is reasonably big, you can still have a very big business.

So to create a valuable company you have to basically both create something of value and capture some fraction of the value of what you've created. And sort of just to illustrate this as a contrast, if you compare the US airline industry with a company like Google on search, if you measure by the size of these industries you could say that airlines are still more important than search, just measured by revenue. [For airline carriers] there's $195 billion in domestic revenues in 2012; Google had just north of $50 billion. And certainly on some intuitive level if you were given a choice and said, well would you want to get rid of all air travel, or do you want to give up search engines, the intuition would be that air travel is something that's more important than search. And this is of course just the domestic numbers.

If you'd look at this globally, airlines are much much bigger than search, than Google is, but the profit margins are quite a bit less. They were marginally profitable in 2012, I think in the entire hundred year history of the airline industry, the cumulative profits in the US have been approximately zero. The companies make money, they episodically go bankrupt, they get recapitalized, and you sort of cycle and repeat. And this is reflected in the combined market capitalization of the airline industry, which may be something like a quarter of Google's. So you have search much smaller than air travel but much more valuable. I think this reflects these very different valuations on "X" and "Y."

If we look at perfect competition, there's some pros and cons to the world of perfect competition. On a high level, this is something you always study in Econ I, it's always easy to model, which is why I think econ professors like talking about perfect competition. It somehow is efficient, especially in a world where things are static, because you have all the consumer surplus that's captured by everybody, and politically it's what we're told is good in our society: you want to have competition and it's somehow a good thing. Of course there are a lot of negatives, it's generally not that good if you're involved in anything that's hyper competitive, because you often don't make money. I'll come back to this a little bit later. So I think on one end of the spectrum you have industries that are perfectly competitive and at the other end of the spectrum you have things that I would say are monopolies, and they're much more stable longer term businesses, you have more capital, and if you get a creative monopoly for inventing something new, I think it's symptomatic of having created something really valuable.

I do think the extreme binary view of the world I always articulate is that there are exactly two kinds of businesses in this world, there are businesses that are perfectly competitive and there are businesses that are monopolies. There is shockingly little that is in between. And this dichotomy is not understood very well because people are constantly lying about the nature of the businesses they are in. And in my mind this is not necessarily the most important thing in business, but I think it's the most important business idea that people don't understand, that there are just these two kinds of businesses.

So let me tell you a little bit about the lies people tell. If you imagine that there was a spectrum of companies from perfect competition to monopoly, the apparent differences are quite small because the people who have monopolies pretend not to. They will basically say, and it's because you don't want to get regulated by the government, you don't want the government to come after you, so you will never say that you have a monopoly. So anyone who has a monopoly will pretend that they are in incredible competition; and on the other end of the spectrum if you are incredibly competitive, and if you're in some sort of business where you will never make any money, you'll be tempted to tell a lie that goes in the other direction, where you will say that you're doing something unique that is somehow less competitive than it looks because you'll want to differentiate, you will want to try and attract capital, or something like that. So if the monopolists pretend not to have monopolies, the non-monopolies pretend to have monopolies, the apparent difference is very small whereas the real difference I would submit is actually quite big. So there's this business distortion that happens because of the lies people tell about their businesses and the lies are sort of in these opposite directions.

Let me drill a little bit down further on the way these lies work. And so the basic lie you tell as a non-monopoly is that we're in a very small market. The basic lie you tell as a monopoly is the market you're in is much bigger than it looks. So typically if you want to think of this in set theoretic terms, you could say that a monopoly tells a lie where you describe your business as the union of these vastly different markets and the non-monopolist describes it as the intersection. So in effect, if you're a non-monopolist you will rhetorically describe your market as super small, you're the only person in that market. If you have a monopoly you'll describe it as super big and there's lots of competition in it.

Some examples of how this works in practice. I always use restaurants as the example of a terrible business, this is always sort of the idea that capital [accumulation] and competition are antonyms. If someone accumulates capital, a world of perfect competition is a world where all the capital gets competed away. So you're opening a restaurant business, no one wants to invest because you just lose money, so you have to tell some idiosyncratic narrative and you'll say something like. "Well we're the only British food restaurant in Palo Alto." So its British, Palo Alto and of course that's too small a market because people may be able to drive all the way to Mountain View or even Menlo Park and there probably are no people who eat nothing but British food, at least no people who are still alive.

So that's sort of a fictitiously narrow market. There is sort of a Hollywood version of all of this, the way movies always get pitched, where it's like a college football star, you know, joins an elite group of hackers to catch the shark that killed his friend. Now that's a movie that has not yet been made, but the question is, "Is that the right category or is the correct category, it's just another movie?" In which case there are lots of those, it's super competitive, it's incredibly hard to make money, no one ever makes money in Hollywood making movies, it's really really hard.

So you always have this question about is the intersection real? Does it make sense? Does it have value? And of course there are startup versions of this, and in the really bad versions you just take a whole series of the buzzwords: sharing, mobile, social apps, you combine them and give some kind of narrative and whether or not that's a real business or not, it is generally a bad sign. It's almost this pattern recognition when you have this rhetoric of these sort of intersections, it generally does not work. The something of somewhere is really mostly just the nothing of nowhere, it's like the Stanford of North Dakota, one of a kind, but it's not Stanford.

Let's look at the opposite, the opposite lie, is if you are let's say the search company that's down the street from here and has about a happy sixty-six percent market share and is completely dominant in the search market. Google almost never describes itself as a search engine these days and instead it describes itself in all these different ways. So it sometimes says it's an advertising company. So if it was search you would say, well it's like it has this huge market share that's really crazy, so it's like an incredible monopoly, it's a much bigger and much more robust monopoly than Microsoft ever had in the nineties, maybe that's why it's making so much money. But if you say it's an advertising market, you could say well, search advertising is seventeen billion and that's part of online advertising, which is much bigger and then, you know, all US advertising is bigger, and then by the time you get to global advertising, that's close to five hundred billion and so you're talking about three and a half percent, so a tiny part of this much larger market.

Or if you don't want to be an advertising company, you could always say you're a technology company. The technology market is something like a one trillion dollar market and the narrative that you tell about Google and the technology market is, well we're competing with all the car companies with our self-driving cars, we’re competing with Apple on TVs and iPhones, we’re competing with Facebook, we’re competing with Microsoft on office products, we’re competing with Amazon on cloud services and so we're in this giant technology market, where there's competition in every direction you look and no we’re not the monopoly the government's looking for and we should not get regulated in any way whatsoever. So I think one has to always be super aware that there are these very powerful incentives to distort the nature of these markets, one way or the other.

The evidence of narrow markets in the tech industry is if you basically just, if you look at sort of the big tech companies, Apple, Google, Microsoft, Amazon, they have just been building up cash for year after year and you have these incredibly high profit margins, and I would say that the that one of the reasons the tech industry in the US has been so successful financially is because it's prone to creating all these monopoly-like businesses and it's reflected by the fact that these companies just accumulate so much cash they don't even know what to do with it beyond a certain point.

Let me say a few things about how to build a monopoly, and I think one of the sort of very counterintuitive ideas that comes out of this monopoly thread is that you want to go after small markets. If you’re a startup, you want to get to monopoly. You’re starting a new company, you want to get to monopoly. Monopolies have a large share of the market, how do you get to a large share of the market? You start with a really small market and you take over the whole market and then over time you find ways to expand that market in concentric circles.

The thing that's always a big mistake is going after a giant market on day one because that's typically evidence that you somehow haven't defined categories correctly, that normally means there is going to be too much competition in one way or another, so I think almost all the successful companies in Silicon Valley had some model of starting with small markets and expanding. If you take Amazon, you start with just a bookstore, we have all the books in the world, it's a better bookstore than anybody else has in the world when it starts in the 1990s. It's online, there's things you can do that you could not do before, and then you gradually expand into all sorts of different forms of e-commerce and other things beyond that.

eBay, you start with Pez dispensers, you move on to Beanie Babies, and eventually it’s all these different online auctions for all these sorts of different goods. What’s very counterintuitive about many of these companies is they often start with markets that are so small, that people don't think they are valuable at all when you get started. The PayPal version of this was we started with power sellers on Ebay, which was about twenty thousand people. When we first saw this happening in December of 1999, January 2000 right after we launched, there was a sense that these were all, it was such a small market, it was terrible, we thought these were terrible customers to have, it's just people selling junk on the internet, why in the world we want to be going after this market?

But there was a way to get a product that was much better for everybody in that market, and we got to something like twenty five, thirty percent market penetration in two or three months, and you've got some walk in to brand recognition, and are able to build the business from there. So I always think these very small markets are quite underrated. The Facebook version of this I always give is that if the initial market at Facebook was ten thousand people at Harvard, it went from zero to sixty percent market share in ten days, that was a very auspicious start. The way this gets analyzed in business schools is always, that's so ridiculous, it's such a small market, it can't have any value at all. So I think the business school analysis of Facebook early on, or of Paypal early on, or of Ebay early on, is that the markets were perhaps so small as to have almost no value and that they would've had little value had they they stayed small, but it turned out there were ways to grow them concentrically and that's what made them so valuable.

Now I think the opposite version of this is always where you have super big markets and there is so many different things that went wrong with all the clean tech companies in the last decade. But the one theme than ran through almost all of them is that they all started with massive markets. Every clean tech powerpoint presentation that one saw in the years 2005 to 2008, which is the clean tech bubble in Silicon Valley, started with where the energy market, where the market was measured in hundreds and billions of trillions of dollars. And then once you're sort of like a minnow in a vast ocean, that is not a good place to be. That means you have tons of competitors and you don’t even know who all the competitors are.

You want to be a one of a kind company. You want to be the only player in a small ecosystem. You don't want to be the fourth online pet food company. You don't want to be the tenth solar panel company. You don't want to be the hundredth restaurant in Palo Alto. Your restaurant industry is a trillion dollar industry. So if you do a market size analysis, you conclude restaurants are fantastic business to go into. And often large existing markets typically means that you have tons of competitions so it's very very hard to differentiate. The first very counterintuitive idea is to go after small markets, markets that are so small people often don't even think that they make sense. That's where you get a foothold and then if those markets are able expand, you can scale into a big monopoly business.

There are several different characteristics of these monopoly businesses that I'd like to focus on. There is no single formula. In technology there is always a sense that in the history of technology, every moment happens only once. The next Mark Zuckerberg won't build a social network and the next Larry Page won't be building a search engine, and the next Bill Gates won't be building an operating system. If you are copying these people, you are not learning from them.

There are always very unique businesses that are doing something that has not been done before and end up having the potential to be a monopoly. The opening line in Anna Karenina, that all happy families are alike and all unhappy families are unhappy in their own special way, is not true in business, where I think all happy companies are different because they're doing something very unique. All unhappy companies are alike because they failed to escape the essential sameness in competition.

One characteristic of a monopoly technology company is some sort of proprietary technology. My sort of crazy, somewhat arbitrary rule of thumb is you want to have a technology that's an order of magnitude better than the next best thing. So Amazon had over ten times as many books, it may not have been that high tech, but you figure oh well it can sell ten times as many books and be more efficient along the way. In the case of PayPal, Bill Turner was using checks to send money on Ebay, it took seven to ten days to clear, and PayPal could do it more than ten times as fast. You want to have some sort of very powerful improvement, some order of magnitude improvement, on some key dimension. Of course, if you just come with something totally new, it's just like an infinite improvement. I would say the iPhone was the first smart phone that worked, it may not be in fact, but it's so definitely an order of magnitude of an improvement. So I think the technology needs to be designed to give you a massive delta over the next the next best thing.

I think there often are network effects that can kick in and that really help the thing and these are linked to monopolies over time, the thing that is very tricky about network effects is they're often very hard to get started and so you know everyone understands how valuable they are. There's always this incredibly tricky question: why is it valuable to the first person who's doing something. Economies of scale, if you have something with very high fixed costs and very low marginal costs, that's typically a monopoly-like business.

And then there is this thing of branding which is sort of this idea that gets lodged into people's brains. I never quite understand how branding works, so I never invest in companies where it's just about branding but it is, I think, a real phenomenon that creates real value. I think one of the things, I'm going to come back to this in a little bit, towards the end, but one of the things that's very striking is that software businesses are often, are for some reason, very good at some of these things. They are especially good at the economies of scale part because the marginal cost of software is zero. So if you get something that works in software it's often significantly better than the existing solution and then you have these tremendous economies of scale and you can scale fairly quickly.

So even if the market starts small, you can grow your business quickly enough to stay at the same size as the growing market and maintain the monopoly of power. Now the critical thing about these monopolies is it's not enough to have a monopoly for just a moment. The critical thing is have one that lasts over time and so in Silicon Valley there is always this sort of idea that you want to be the first mover and I always think in some ways the better framing is you want to be the last mover. You want to be one of the last companies in that category, those are the ones that are really valid. Microsoft was the last operating system, at least for many decades. Google was the last search engine. Facebook will be valuable if it turns out to be the last of social networking site.

And one way to think of this last mover value is this idea that most of the value in these companies exists far in the future. If you do a discounted cash flow analysis of the business, you'll look at all these profit streams, you have a growth rate, the growth rate's much higher that the discount rate and so most of the value exists far in the future. I did this exercise at

Pay Pal in March of 2001 we'd been in business for about twenty seven months and the growth rate was a hundred percent a year, we were discounting future cash flows by thirty percent, and it turned out that about three quarters of the value of the business as of 2001 came from cash flows in years 2011 and beyond.

And whenever you do the math on any of these tech companies, you get to an answer that is something like that. So if you are trying to analyze any of the tech companies in Silicon Valley, AirBnB, Twitter, Facebook, any emerging Internet companies, all the ones in Y Combinator, the math tells you that three quarters, eighty-five percent of the value is coming from cash flows in years 2024 and beyond. It's very far in the future and so one of the things that we always over value in Silicon Valley is growth rates and we undervalue durability.

Growth is something you can measure in the here and now, you can always track that very precisely. The question of whether a company will be around a decade from now, that's actually what dominates the value equation and that's a much more qualitative sort of a thing. And so if we went back to this idea of these characteristics of monopoly, the proprietary technology, network effects, economies of scale, you can think of these characteristics as ones that exist at a moment in time where you capture a market and take it over but you also want to think about, are these things going to last over time. So there's a time dimension to all these characteristics. So networks in fact often have a great time element where as the network scales, network effects actually get more robust, and so if you have a network business it's often one that can become a bigger and stronger monopoly over time.

Proprietary technology is always a little bit of a tricky one, so you want something that is an order of magnitude better than the state of the art in the world today. That's how you get people's attention, that's how you initially break through. But then you don't want to be superseded by somebody else. So there are all these areas of innovation where there was tremendous innovation, but no one made any money. So disk drive manufacturing in the 1980's, you could build a better disk drive than anybody else, you could take over the whole world and two years later someone else would come along and replace yours. In the course of fifteen years, you got vastly improved disk drives, so had a great benefit to consumers, but it didn't actually help the people who started these companies.

There's always this question about having a huge breakthrough in technology, but then also being able to explain why yours will be the last breakthrough or at least the last breakthrough for a long time or if you make a breakthrough, then you can keep improving on it at quick enough pace that no one can ever catch up. So if you have a structure of the future where there's a lot of innovation and other people will come up with new things and the thing you're working on, that's great for society. It's actually not that good for your business typically. And then economies of scale we've already about. I think this last mover thing is very critical. I'm always tempted, I don't want to overdo chess analogies, but the first mover in chess is someone who plays white, white is about a one-third of a pawn advantage, so there is a small advantage to going first. You want to be the last mover who wins the game, so there's always world chess champion Capablanca line, "You must begin by studying the end game." I wouldn't say that's the only thing you should study, I think perspective of asking these questions, why will this still be the leading company in ten, fifteen, twenty years from now, is a really critical one to try to think through.

I want to go in two slightly other directions with this the monopoly versus competition idea. I think this is the central idea on my mind for business, for starting a business, for thinking about them. There are some very interesting perspectives, I think it gives, on the whole history of innovation in technology and science. We've lived through 300 years of incredible technological progress in many many different domains. Steam engines to railways, the telephone, refrigeration, household appliances, the computer revolution, aviation, all different areas of technological innovation. Then there's sort of an analogous thing to say about science where we've lived through centuries of enormous amounts of innovation in science as well.

The thing that I think people always miss when they think about these things, is that because "X" and "Y" are independent variables, some of these things can be extremely valuable innovations, but the people who invent them, who come up with them, do not get rewarded for this. Certainly if you go back to you need to create X dollars in value and you capture Y percent of X, I would suggest that the history of science has generally been one where Y is zero percent across the board, the scientists never make any money. They're always deluded into thinking that they live in a just universe that will reward them for their work and for their inventions. This is probably the fundamental delusion that scientists tend to suffer from in our society. Even in technology there are sort of many different areas of technology where there were great innovations that created tremendous value for society, but people did not actually capture much of the value. So I think there is a whole history of science and technology that can be told from the perspective of how much value was actually captured. Certainly there are entire sectors where people didn't capture anything.

You're the smartest physicist of the twentieth century, you come up with special relativity, you come up with general relativity, you don't get to be a billionaire, you don't even get to be a millionaire. It just somehow doesn't work that way. The railroads were incredibly valuable, they mostly just went bankrupt because there was too much competition. Wright brothers, you fly the first plane, you don't make any money. So I think there is a structure to these industries that’s very important.

I think the thing that's actually rare are the success cases. So if you really think about the history in this and this two hundred fifty years sweep, why is almost always zero percent, it's always zero in science, it's almost always in technology. It's very rare where people made money. You know in the late eighteenth, early nineteenth century, the first industrial revolution was the textile mills, you got the steam engine, you sort of automated things. You had these relentless improvements that people improved efficiency of textile factories, of manufacturing generally, at a clip of five to seven percent every year, year after year, decade after decade. You had sixty, seventy years of tremendous improvement from 1780 to 1850. Even in 1850, most of the wealth in Britain was still held by the landed aristocracy and the workers didn't make that much. The capitalists didn't make that much either, it was all competed away. There were hundreds of people running textile factories, it was an industry where the structure of the competition prevented people from making any money.

There are, in my mind, probably only two broad categories in the entire history of the last two hundred and fifty years where people actually came up with new things and made money doing so. One is these sort of vertically integrated complex monopolies which people did build in the second industrial revolution at the end of the nineteenth and start of the twentieth century. This is like Ford, it was the vertically integrated oil companies like Standard Oil, and what these vertically integrated monopolies typically required was a very complex coordination, you've got a lot of pieces to fit together in just the right way, and when you assemble that you had a tremendous advantage. This is actually done surprisingly little today and so I think this is sort of a business form that when people can pull it off, is very valuable.

It's typically fairly capital intensive, we live in a culture where it's very hard to get people to buy into anything that's super complicated and takes very long to build. When I think of my colleague and friend Elon from PayPal success with Tesla and SpaceX, I think the key to these companies was the complex vertically integrated monopoly structure they had. If you look at Tesla or SpaceX and you ask, was there sort of a single breakthrough, I mean they certainly innovated on a law of dimensions, but I don't think there was a single 10X breakthrough on battery storage, they may be working on some things in rocketry, but there was no sort of single massive breakthrough. But what was really impressive was integrating all these pieces together and doing it in a way that was more vertically integrated than most other competitors.

So Tesla also integrated the car distributors so they wouldn't steal all the money as has happened with the rest of the car industry in the US. Or SpaceX, basically, you pulled in all subcontractors where most of the large aerospace companies have single sourced subcontractors that are able to charge monopoly profits and make it very hard for the integrated aerospace companies to make money. And so vertical integration I think is sort of a very under explored modality of technological progress that people would do well to look at more.

And then I think there is something about software itself that's very powerful. Software has these incredible economies of scale, these low marginal costs, and there is something about the world of bits, as opposed to the world of atoms, where you can often get very fast adoption and fast adoption is critical to capturing and taking over markets because even if you have a small market, if adoption rate is too slow, there will be enough time for other people to enter that market and compete with you. Whereas if you have a small to midsize market, have the fast adoption rate, you can now take over this market. So I think this is one of the reasons Silicon Valley has done so well and why software has been this phenomenal industry.

What I would suggest that we will leave you with is there are these different rationalizations people give for why certain things work and why certain things don't work, and I think these rationalizations always obscure this question on creating "X" dollars in value and capturing "Y" percent of "X." So, the science rationalization we’re always told is that the scientists aren't interested in making money. They're doing it for charitable reasons and that you're not a good scientist if you're motivated by money. I'm not even saying people should always be motivated by money or something like this, but I think we should wish to be a little bit more critical of this as a rationalization. We should ask is this a rationalization to obscure the fact that "Y" equals zero percent and the scientists are operating in this sort of world where all the innovation is effectively competed away and they can't capture any of it directly.

The software distortion that often happens is because people are making such vast fortunes in software, we infer that this is the most valuable thing in the world being done full stop. And so people at Twitter make billions of dollars, it must be that Twitter is worth far more than anything Einstein did. What that realization tends to obscure, is again that "X" and "Y" are independent variables and that there are these businesses where you capture a lot of X and others where you don't. So I do think the history of innovation has been this history where the microeconomics, the structure of these industries has mattered a tremendous amount and there is sort of this story where some people made vast fortunes because they worked in industries with the right structure and other people made nothing at all because they were in these very competitive things.

We shouldn't just rationalize that way. I think it's worth understanding this better. Then finally, let me come back to this sort of overarching theme for this talk, this competition is for losers idea, which is always a provocative way to title things because we always think of the losers as the people who are not good at competing. We think of losers as the people who are slow on the track team in high school or do a little less well on standardized tests, and don't get into the right schools. So we always think of losers as people who can't compete and I want us to really rethink and re-value this and consider whether it's possible the competition itself is off.

It's not just the case we don't understand this monopoly-competition dichotomy intellectually. So we've been talking about why you wouldn't understand it intellectually, because people lie about it, it's distorted, the history of innovation rationalizes it in all these very strange ways. I think it's more than just an intellectual blind spot, but also a psychological blind spot, where we find ourselves very attracted to competition and in one form or another we find it reassuring if other people do things. The word ape, already in the time of Shakespeare, meant both primate and imitate and that is something about human nature, it's deeply imitative, ape like, sheep like and this is very problematic thing that we need to always think through and try to overcome.

There is always this question about competition as a form of validation, where we go for things that lots of other people are going for. It's not that there is wisdom of crowds, it's not that lots of people trying to do something is the best proof of that being valuable. I think it's when lots of people are trying to do something, that is often proof of insanity. There are twenty thousand people a year who move to Los Angeles to become movie stars, about twenty of them make it. I think the Olympics are a little bit better because you have, you can sort of figure out pretty quickly whether you're good or not, so there's little less of a deadweight loss to society. You know the sort of educational experience that at a place, the pre-Stanford educational experience, there is always sort of a non-competitive characterization. I think most of the people in this room had machine guns and they were competing with people with bows and arrows, so it wasn't exactly a parallel competition when you were in junior high school, in high school. There is always the question: does the tournament make sense as you keep going?

There is always this question if people going on to grad school or post doctoral educations, does the intensity of the competition really make sense. There is the classic Henry Kissinger line describing his fellow faculty at Harvard, "The battles were so ferocious because the stakes were so small," describing academia and you sort of think on one level this is a description of insanity. Why would people fight like crazy when the stakes are so small, but it's also, I think, simply a function of the logic of a situation. When it's been really hard to differentiate yourself from other people, when the differences are, when the objectives differences really are small, you have to compete ferociously to maintain a difference of one sort or another. That's often more imaginary than real. There is always sort of a personal version of this I tell, where I was sort of hypertracked. In my eighth grade junior high school yearbook one my friends wrote, I know you're going to get into Stanford. Four years later, I went to Stanford Law School, ended up at a big law firm in New York where from the outside everybody wanted to get in and on the inside everybody wanted to leave and it was this very strange dynamic where I realized, this was maybe not the best idea, and I left after seven months and three days.

Other people down the hall told me, it's really reassuring to see you leave, Peter, I had no idea that it was possible to escape from Alcatraz, which of course all you had do was go out the front door and not come back. But so much of people's identities got wrapped up in winning these competitions that they somehow lost sight of what was important, what was valuable. Competition does make you better at whatever it is that you're competing at because when you're competing you're comparing yourself with the people around you. I'm figuring out how to beat the people next to me, how do I do somewhat better than whatever it is they're doing and you will get better at that. I'm not questioning that, I'm not denying that, but there often comes this tremendous price that you stop asking some bigger questions about what's truly important and truly valuable. Don't always go through the tiny little door that everyone's trying to rush through, maybe go around the corner and go through the vast gate that nobody is taking.

Q: Do you have any ways to determine the difference [between a monopoly and a non monopoly] when looking at an idea or thinking about your own idea?

A: I would say the question I always focus on is what is the actual market? So not what's the narrative of the market, because you can always tell a fictional story about a market: it's much bigger much or smaller, but what is the real objective market. So you always try to figure it out, and you realize people have incentives to powerfully distort these things.

Q: So which of the aspects, of all these you mentioned, you would you say is applicable to Google?

A: Well, they have they have network effects with the ad network, they had proprietary technology that gave them the initial lead because they had the page rank algorithm, which was an order of magnitude better than any other search engine. They had economies of scale up because of the need to store all these different sites, and at this point you have brand, so Google has all four. Maybe the proprietary technology is somewhat weaker at this point but definitely it had all four, and maybe three and a half out of four now.

Q: How does this apply to Palantir and Square?

A: That's sort of a set of companies that are doing different copycat payment systems, on mobile phones, there's Square, there's PayPal, they have different shapes that's how they differentiate themselves, one is a triangle, one is a square. Maybe at one point the apes run out of shapes or something like that, but at Palantir we started with a focus on the intelligence community, which is a small submarket. We had a proprietary technology that used a very different approach where it was focused on the human computer synthesis, rather than the substitution, which I think is the dominant paradigm. So, there is a whole set of things, I would say, on the market approach and on the proprietary technology.

Q: What do you think about lean startups?

A: So the question is what do I think about lean startups and iterative thinking where you get feedback from people versus complexity that may not work.

I'm personally quite skeptical of all the lean startup methodology. I think the really great companies did something that was somewhat more of a quantum improvement that really differentiated them from everybody else. They typically did not do massive customer surveys, the people who ran these companies sometimes, not always, suffered from mild forms of Aspergers, so they were not actually that influenced, not that easily deterred, by what other people told them to do. I do think we're way too focused on iteration as a modality and not enough on trying to have a virtual ESP link with the public and figuring it out ourselves.

I would say the risk question is always a very tricky one, because it's often the case that you don't have enough time to really mitigate risk. If you're going to take enough time to figure out what people want, you often will have missed the boat by then. And then of course there is always the risk of doing something that's not that significant or meaningful. You could say that a track in law school is a low risk track from one perspective, but it may still be a very high risk track in the sense that maybe you have a high risk of not doing something meaningful with your life. We have to think about risk in these very complicated ways. I think risk is this complicated concept.

Q: Doesn't the last mover advantage already imply that there's competition to begin with?

A: Yes, there's always a terminology thing. I would say that there are categories in which people sort of are bundled together. The monopoly business, I think they really were a big first mover. In some sense you can say Google was not the first search engine, there were search engines before. But on one dimension they were dramatically better than everybody else. They were the first one with page rank, with an automated approach. Facebook was not the first of social networking site. My friend, Reid Hoffman, started one in 1997, they called it Social Net, so they already had the name social networking in the name of their company seven years before Facebook. Their idea was that it going to be this virtual cyberspace were I'd be a dog and you'd be a cat and we'd have all these different rules about how we interact with each other in this virtual alternate reality. Facebook was the first one to get real identity, so I hope Facebook will be the last social networking site. It was the first one in a very important dimension, people often would not think of it as the first because they sort of lump all these things together.

Q: If someone worked at Goldman Sachs out of college and left after six months and is now studying CS at Stanford, how would you recommend rethinking their competitive advantage?

A: I am not great at the psychotherapy stuff, so I don't quite know how to solve this. There are these very odd studies they have done on people who go to business school, this one was done at the Harvard Business School where it's sort of the anti-Asbergers personality, where people are super extroverted, generally have low convictions, few ideas and you have sort of a hothouse environment you put all these people and for two years and at the end of it, they systematically end up, the largest cohort systematically ends up doing the wrong thing, they try to catch the last wave. in 1999 everyone tried to work with Mike Milken, this was a few years before he went to jail for all the junk bond stuff.

They were never interested in Silicon Valley or tech except for 1999, 2000 when they timed the dotcom bubble peaking perfectly. 2005 to 2007 was housing, private equity, stuff like this. This tendency for us to see competition as validation is very deep, I don't think there's some easy psychological formula to avoid this. I don't quite know how what sort of therapy to recommend.

My first starting point, which is only going to be maybe ten percent of the way, is to never underestimate how big a problem it is. We always think that this is something that afflicts other people. We always point to people in business school, people at Harvard or people on Wall Street, but it actually does afflict all of us to a very profound degree. We always think of advertising as this thing that works on other people, for all the stupid people who follow ads on TV, but they obviously work to some extent and they work to the disturbing extent on all of us and it's something we must work to overcome.

Thank you very much.

Lecture 6: Growth

Alex Schultz

This is awesome; I've been watching the lectures on this course, and isn't it absolutely amazing, the content? And now, you're stuck with me. We'll see how that goes.

Unlike Paul when he was talking in the Q&A and you guys asked him what he'd do if he was in college today and he said physics, I actually indulged myself. I went and did physics at Cambridge. I think physics is an amazing class to give you transferable skills that are really useful in other areas, but that's not why you're listening to me today; physics isn't the class.

So I paid for college doing online marketing, directions marketing. I started with SEO in the 1990s. I created a paper airplane site, and had a monopoly in the small niche market of paper airplanes. When you want to start a startup also see how big the market could be. (In the long term, it wasn't great.) But what that taught me was how to do SEO. And back in those days it was Alta Vista, and the way to do SEO was to have white text, on a white background, five pages below the fold, and you would rank top of Alta Vista if you just said planes 20 or 30 times in that text. And that was how you won SEO in the 1990s. It was a really, really easy skill to learn.

When I went to college, being a physicist, I thought paper airplanes would make me cool. I was actually the nerdiest person in the physics class, so I created a cocktail site, which was how I learned to program and that grew to be the largest cocktail site in the UK. That really got me into SEO properly when Google launched. So with Google you had to worry about page rank and getting links back to your site, which basically at that stage meant one link from the Yahoo directory, got you to the top listing in Google if you had white text and a white background below the fold as well.

When Google launched AdWords, that's really when I started to do all my marketing. That meant buying paid clicks from Google and reselling them to eBay for a small margin of like 20% using their affiliate program. That was what really kicked me into overdrive, into doing what everyone nowadays talks about as growth, growth hacking or growth marketing. In my mind it's just internet marketing using whatever channel you can to get whatever output you want, and that's how I paid for college and that's how I went from being a physicist to a Marketer - transitioning to the darkside of the force.

So what do you think matters most for growth? You've had tons of lectures, and people have said it over and over, so what do you guys think matters most for growth?

Audience: Great product.

Schultz: What does a great product lead to?

Audience: Customers.

Schultz: And what do you need those customers to do?

Audience: Spread the word.

Schultz: Yes that's it, retention. Retention is the single most important thing for growth. Now we have an awesome growth team at Facebook and I'm super proud to work in it, but the truth of the matter is, we have a fantastic product. Getting to work on growth of Facebook is a massive privilege because we are promoting something that everyone in the world really wants to use, which is absolutely incredible. If we can get people on, and get them ramped up, they stick on Facebook.

So many times, I got to advise multiple startups. My favorite was working with Airbnb, but I've worked with Coursera, I've worked with other ones that haven't done as well as those guys. But the one thing that's true, over and over again is, if you look at this curve, ‘percent monthly active’ versus ‘number of days from acquisition’, if you end up with a retention curve that is asymptotic to a line parallel to the X-axis, you have a viable business and you have product market fit for some subset of market. But most of the companies that you see fly up, we’ve talked about packing and virality and all of this stuff, their retention curve slopes down toward the axis, and in the end, intercepts the X-axis.

Now when I show this job to people, they say that’s all well and good, you had a million people a day in terms of growth, when you started the growth team at Facebook, or ‘you were at 50 million users, you had a lot of people join your site so you had a ton of data to do this.’ We used the same methodology for our B2B growth , getting people to sign-up for services advertisements, we used this to understand how much growth we were going to have in that market as well. And at that point when I joined Facebook, the product was three days old. And within 90 days of the product launching, we were able to use this technique to figure out what the one year value of an advertiser was, and we predicted it for the first year to 97%. So I think it’s very important to look at your retention curve.

If you see here, this red line is the ‘number of users’ who have been on your product for a certain number of days. So a bunch of people, will have been on the product at least one day, but if your product has been around for a year, you’ll have zero users who have been on it for 366 days. Make sense?

So what you then do is look for all of your users who have been on your product one day. What percentage of them are monthly active? 100% for the first 30 days obviously, because monthly active, they also end up on one day. But then you look at 31. Every single user on their 31st day after registration, what percentage of them are monthly active? Thirty-second day, thirty-third day, thirty-fourth day. And that allows you, with only 10,000 customers, to get a real idea of what this curve is going to look like for your product. And you’re going to be able to tell, is it asymptotic? It’ll get noisy towards the right side, like I’ m not using real data, but you’ll be able to get a handle on, whether this curve flattens out or does it not. If it doesn’t flatten out, don’t go into growth tactics, don’t do virality, don’t hire a growth hacker. Focus on getting product market fit, because in the end, as Sam said in the beginning of this course: idea, product, team, execution. If you don’t have a great product, there’s no point in executing more on growing it because it won’t grow. Number one problem I’ve seen, inside Facebook for new products, number one problem I’ve seen for startups, is they don’t actually have product market fit, when they think they do.

So the next question that people ask over and over again is, what does good retention look like? Sure! I can have 5% retention, but I’m guessing Facebook had better than that. That’s not going to be a successful business. I get really pissed off when people ask me that question, because I think you can figure it out. I love this story; this is like my one gratuitous story (link on powerpoint) that I’m throwing out here, so the rest of it may not be as gratuitous. But this is a picture that was published in Life Magazine in 1950 of one of the Trinity nuclear bomb tests. There’s a guy named Jeffrey Taylor. He was a British Physicist who ended up winning the Nobel Prize. He was able to figure out, from looking at this picture (picture on powerpoint) what the power of the U.S. atomic bomb was, and Russians were publishing similar pictures, just using dimensional reasoning. Dimensional reasoning was one of the best skills I learned during my time studying physics back in the UK.

What dimensional reasoning is, you look at the dimensions that are involved in a problem, so you want to figure out energy, newtons, meters, newtons as a kilogram, meters seconds to minus two. You want to figure out kilograms, meters squared, seconds to minus two, and then you try to figure out how you can get each of those numbers from what data you have. The mass is the volume of this sphere, so that’s a meter cubed, so you’ve got meters to five over seconds to minus two and he was able to use that to figure out what the power of this atomic bomb was and what the ratios of the power between the Russian and the U.S. atomic bomb was, and essentially reveal one of the top secrets that existed in the world at that time.

That’s a hard problem. Figuring out what Facebook’s retention rate is, is not a hard problem. How many people are there on the internet? 2.4 billion, 2.3 billion. Okay, Facebook is banned in China, so what now?

Audience: 2 billion.

Schultz: So 2 billion people on the internet. Facebook said around 1.3 billion users in terms of active users. You can divide those numbers by each other. And yet that won’t give you the right answer. Of course not! But it’s going to give you close enough to a ballpark answer of what the retention rate looks like for Facebook. If we signed everyone on the internet up, then you will know it's higher than that. Similarly, if you look at WhatsApp. They've announced 600 million active users. How many people have Smart Phones? You can figure out that number - that number is out knocking around. It can give you an idea of how many users there are. Amazon has a had a pop at signing up almost everyone in the United States. You know how many people are online in the United States, and you have a good idea of how many customers Amazon has from the numbers they throw out. Different verticals need different terminal retention rates for them to have successful businesses. If you're on ecommerce and you're retaining on a monthly active basis, like 20 to 30% of your users, you're going to do very well. If you're on social media, and the first batch of people signing up to your product are not like, 80% retained, you're not going to have a massive social media site. So it really depends on the vertical you're in, what the retention rates are. What you need to do is have the tools to think, 'who out there is comparable' and how you can look at it and say, 'am I anywhere close to what real success looks like in this vertical?'

Retention is the single most important thing for growth and retention comes from having a great idea and a great product to back up that idea, and great product market fit. The way we look at, whether a product has great retention or not, is whether or not the users who install it, actually stay on it long-term, when you normalize on a cohort basis, and I think that's a really good methodology for looking at your product and say 'okay the first 100, the first 1,000, the first 10,000 people I get on this, will they be retained in the long-run?

So now, how do you attack operating for growth? Let's say you have awesome product market fit. You've built an ecommerce site, and you have 60% of people coming back every single month, and making a purchase from you, which would be absolutely fantastic. How do you then take that, and say, 'now it's time to scale.' (Now it's time to execute was the last thing on your forum right? *to moderator*.) That's where I think growth teams come in.

My contrarian viewpoint is, if you're a startup, you shouldn't have a growth team. Startups should not have growth teams. The whole company should be the growth team. The CEO should be the head of growth. You need someone to set a North star for you about where the company wants to go, and that person needs to be the person leading the company, from my opinion, that's what I've seen. Mark is a fantastic example of that. Back when Facebook started, a lot of people were putting out their registered user numbers. Right? You'd see you registered user numbers for MySpace, you'd see a registered user numbers for ___11:38, you'd see registered user numbers. Mark put out monthly active users, as the number both internally he held everyone to, and said we need everyone on Facebook, but that means everyone active on Facebook, not everyone signed up on Facebook, so monthly active people was the number internally, and it was also the number he published externally. It was the number he made the whole world hold Facebook to, as a number that we cared about. If you look at what Jan has done with WhatsApp I think that's another great example. He always published sends numbers.

If you're a messaging application, sends is probably the single most important number. If people use you once a day, maybe that's great, but you're not really their primary messaging mechanism, so Jan published the sends number. Inside Airbnb, they talk about 'nights booked' and also published that in all of the infographics you see in side TechCrunch. They always benchmark themselves against how many nights booked they have compared to the largest hotel chains in the world. They have at each of these companies, a different north star. The north star doesn't have to be the number of active users for every different vertical. For eBay, it was gross merchandise volume. How much stuff did people actually buy through eBay? Everyone externally tends to judge eBay based on revenue. Actually, Benedict Evans has done this amazing breakdown of Amazon's business, which is really interesting to look at their marketplace business versus their direct business. So eBay is all marketplace business, right? So eBay's being judged by its revenue, when it actually has 10 times Gross Merchandise Volume going through the site. That was the number that eBay looked at when I was working there. Every different company when it thinks about growth, needs a different North star; however, when you are operating for growth it is critical that you have that North star, and you define as a leader.

The reason this matters is, the second you have more than one person working on something, you cannot control what everyone else is doing. I promise you, having now hit 100 people I'm managing, I have no control. It's all influence. It's like I tell one person to do one thing, but the other 99 are going to do whatever they want. And the thing is, it's not clear to everybody what the most important thing is for a company. It would be very easy for people inside eBay to say, 'you know what? we should focus on revenue,' or 'we should focus on the number of people buying from us' or 'we should focus on how many people list items on eBay.' And Pierre, and Meg, and John, those guys as various leaders, have always said ‘no, its the amount of Gross Merchandise Volume that goes through our site, the percentage of e-commerce that goes through our site, that is what really matters for this company. This means that when someone is having a conversation and you’re not in the room, or when they’re sitting in front of their computer screens, and thinking about how they built this particular project or this particular feature, in their head it’s going to be clear to them that it’s not about revenue, it’s about Gross Merchandise Volume, or it’s not about getting more registrations, registrations don’t matter, unless they become long-term active users. A great example of this was when I was at eBay in 2004, we changed they way we paid our affiliates for new users. Affiliate programs are a bit out of fashion these days, but the idea of an affiliate program is essentially, you pay anyone on the internet a referral for sending traffic to your site, but it’s mostly about getting access to big marketers who do it on their own.

We were paying for confirmed registered users, so all of our affiliates were lined up on getting confirmed registered users to the eBay site. We changed our payment model to pay for activated confirmed registered users. So you had to confirm your account and then bid on an item, or buy or list an item, to become someone that we paid for. Overnight when we made that change, we lost something like 20% of confirmed registered users that were being driven by the affiliates. But the ACRUs (15:45) only dropped by about 5%. The ratio between CRU to ACRU went up, and then, the growth of ACRUs massively accelerated. The cause of this is, if you want to drive CRU, if someone searches for a trampoline, you land them on the registration page because they link you have to register and confirm before they get their trampoline. If you want to drive ACRUs, you land them on the search results page, within eBay for trampolines, so they can see the thing they want to buy, get excited, and then register when they want to buy it. And if you drive just CRUs, people don’t have an amazing magic moment on eBay, when they visit the site. And that’s the next most important thing to think about: How do you drive to the magic moment that gets people hooked on your service.

In the lecture notes for this course, I’ve stuck in a bunch of links to people I think are brilliant at this stuff. For example, regarding the retention curve I showed you earlier, there’s a link to this guy Danny Ferante who is incredible talking about retention curves. The magic moment there are two videos linked: one is Chamath talking about growth, who is the guy who set up the growth team at Facebook, and the other, is my friend Naomi and I talking at f8 four years about how we were thinking about growth back then. In both of those videos, we talk about the magic moment. What do you think the magic moment is for when you’re signing up to Facebook?

Audience: See your friends.

Schultz: See your friends. Simple as that. I’ve talked to so many companies, and they try to get incredibly complicated about what they’re doing, but it is just as simple as when you see the first picture of one of your friends on Facebook, you go ‘Oh my God, this is what this site is about!’ Zuckerberg talked at Y Combinator about getting people to 10 friends in 14 days; that is why we focus on this metric. The number one most important thing in a social media site is connecting to your friends, because without that, you have a completely empty newsfeed, and clearly you’re not going to come back; you’ll never get any notifications, and you’ll never get any friends telling you about things they are missing on the site.

So for Facebook the magic moment, is that moment when you see your friend’s face, and everything we do on growth, if you look at the Linkedin registration flow, you look at the Twitter registration flow, or you look at what WhatsApp does when you sign up, the number one thing all these services look to do, is show you the people you want to follow, connect to, send messages to, as quickly as possible, because in this vertical, this is what matters. When you think about Airbnb or eBay, it’s about finding that unique item, that PEZ dispenser or broken laser pointer, that you really really cared about and want to get ahold of. Like when you see that collectible that you are missing, that is the real magic moment on eBay. When you look on Airbnb and you find that first listing, that cool house you can stay in, and when you go through the door, that’s a magic moment. Similarly on the other side, when you’re listing your house, that first time you get paid, is your magic moment or when you list an item on eBay, the first time you get paid, is your magic moment. You should ask Brian what he thinks, because they’ve done these amazing story boards which I think has been shared, about the journey through a user’s life on Airbnb and how exciting it is. He’ll be talking in around three lectures time; he’s awesome about talking about the magic moment, and getting users to feel the love, joy, and all this stuff.

Think about what the magic moment is for your product, and get people connected to it as fast as possible, because then you can move up where that blue line has asymptotic, and you can go from 60% retention to 70% retention easily if you can connect people with what makes them stick on your site.

The second thing to think about, that everyone in the Valley gets wrong is, we optimize when we think about growth for ourselves. My favorite example is notifications. Again, I've talked to and advised many different companies; every single company when they talk about notifications goes 'Oh, I'm getting too many notifications, I think that's what we have to optimize for on notifications.' Okay, are your power users leaving your site because they're getting too many notifications? No. Then why would you optimize that? They're probably grown-ups and they can use filters.

What you need to focus on is the marginal user. The one person who doesn't get a notification in a given day, month, or year. Building an awesome product is all about think about the power user, right? Building an incredible product is definitely optimizing it for the people who use your product the most, but when it comes to driving growth, people who are already using your product are not the ones you have to worry about. So in this Danny Ferante video there's also talk about our growth accounting framework that we use to think about for growth. We looked at new users, resurrected users (people who weren't on Facebook for 30 days and came back) and churned users. The resurrected and churned numbers for pretty much every product I've ever seen dominate the new user account once you reach a sensible point of growth a few years in. And all those users who are churning and resurrecting, had low friend counts, and didn't find their friends so weren't connected to the great stuff that was going on on Facebook. So the number one thing we needed to focus on, was getting them to those 10 friends, or whatever number of friends they needed. So think about the user on the margin; don't think about where yourself (21:08), when you're thinking about growth.

So for operating for growth, what you really need to think about, is what is the North star of your company: What is that one metric, where if everyone in your company is thinking about it and driving their product towards that metric and their actions towards moving that metric up, you know in the long-run your company will be successful. By the way, they're all probably all correlated to each other, so it's probably fine to pick almost any metric, whichever one you feel the best about, that aligns with your mission and your values - probably go for that one. But realistically, daily active users fairly correlate to monthly active users; we could have gone with either one. Amount of content shared, also correlates with how many users there are, because guess what? You add a user, they share content. So a lot of things end up being correlated. Pick the one that fits with you and know that you're going to stick with for a long time. Just have a North Star, and know the magic moment that you know when a user experiences that, they will deliver on that metric for you on the North Star, and then think about the marginal user, don't think about yourself. Those are, I think, the most important points when operating for growth. Everything has to come from the top.

So the last area is tactics. So let's say you've found your niche market that you're going to have a monopoly on inside the mousetrap market. It's a silenced mousetrap fir sitting under beds, so if that the mice come to your bed overnight, they can be killed without waking you up. That's your niche market. Your mousetrap is better than anybody else for that market. What typically happens in Silicon Valley is, everyone thinks marketers are useless. I thought marketers were useless when I was a Physics student, so I'm sure that you guys as Engineering students you must think that we're awful people who aren't useful to have around.

'Build it and they will come.' That is something that is very much the mantra in the Valley, and I don't believe it's true; I believe you actually have to work. There's a good article in the lecture page from interviewing Ben Silverman. We talked about how the growth of Pinterest was driven by marketing. I'm biased of course.

The first tactic I want to talk about is internationalization. Facebook internationalized too late. Sheryl said it broadly in public and I definitely agree with that.

One of the biggest barriers to our long-term growth, and one of the biggest things we had to deal with, was all the countries where there were clones. Famously ? (23:55) had Fakebook.css in their HTML, and there were a ton of sites like that out there, whether it was ?, a clear clone, Mixie, Cyworld, Orkut; they were all these different social networks around the world that grew up when Facebook was focused around the U.S. Internationalizing was an important barrier we needed to knock down, and knocking down barriers is often important to think about for growth. Facebook started out as college-only, so every college that it was launched in was knocking down a barrier. When Facebook expanded beyond colleges to high schools, I wasn't at the company, but that was a company-shaking moment where people questioned whether or not Facebook would survive,if the culture of the site could survive.

Then after, expanding from high schools to everyone - that was just before I joined - it was a shocking moment; that's what spurred the growth up to 50 million, and then we hit a brick wall. When we hit that brick wall, that was when a lot of existential questions were being asked inside Facebook whether any social network could ever get to more than 100 million users. It sounds stupid now, but at that time, no one had ever achieved it. Everyone had tapped out between 50 and 100 million users, and we were worried that it wasn't possible. That was the point at which the growth team got set up; Chamath brought a bunch of us together. He said very publicly he wanted to fire me on multiple occasions. Without Chamath, I think none of us would have stayed at the company; we were a really weird bunch of people - but it worked out. The two things we did, I think that really drove growth initially was, 1) We focused on that 10 friends in 14 days 2) Getting users to the magic moment. That was something that Zuck drove because we were all stuck in analysis paralysis and, 'Is it causation? Is it correlation?' Zuck would say 'You really think that if no one gets a friend, that they'll be active on Facebook? Are you crazy?'

The second thing was internationalization - knocking down another barrier. When we launched it, I think there were two things we did really well: 1) Even though we were late (and stressed about being late) we took the time to build it in a scalable way; we moved slow to move fast. You can actually view the full story from Naomi on one of the video links from the lecture page. What we did was draw all the strings on the site in FBT, which is our translation extraction script and then, we created the community translation platform, so we didn't have just professional translators translating the site, but we could have all our users translating the site. We got French translated in 12 hours. We managed to get, to this day, 104 languages translated by Facebook for Facebook, 70 of those are translated by the community. We took the time to build something, that would enable us to scale.

The other thing is that we prioritized the right languages. Back then, the four main languages were French, Italian, German, and Spanish (and Chinese, but we are blocked in China). Now look at that list - that's today's distribution of languages. Italian isn't on the list anymore; French and German are about to fall off. In the last year we quadrupled the number of people on Facebook in Hindi. Building for what the world is today is an easy mistake to make, and it's a lot of what the other social networks did. We built a scalable translation infrastructure that actually enabled us to attack all of the languages, so we could be ready for where the future is going to be. You'll probably be able to see some of our Internet.org summit in India about where we want to go with language translations.

These are the tactics I want to go through now: Virality, SEO, ESPN, SEM, Affiliates/referral programs. I think there are two ways to look at virality. There's a great book by Adam L. Penenberg called the Viral Loop that goes through a bunch of case studies of companies that have grown through viral marketing. I strongly encourage you to read this book if you're interested in viral marketing, as well as advertising. I think Ogilvy on Advertising is great as well because in the chapter 7 you can't think of anything else stick a car to billboard with super glue and people will buy the super glue. He has some really great creative tips. So virality. Sean Parker has this really great model that he told us about when I joined Facebook, which is to think about virality about a product, in terms of three things. First, is payload - so how many people can you hit with any given viral blast. Second, is conversion rate, and third is frequency. This gives you a fundamental idea of how viral a product is.

Hotmail is the canonical example of brilliant viral marketing. Back when Hotmail launched, there were a bunch of mail companies that had been funded and were throwing huge amounts of money at traditional advertising. Back in that time, people couldn't get free email clients; they had to be tied to their ISP. Hotmail and a couple other companies launched, and their clients were available wherever you went. You could log-in via library internet or school internet, and be able to get access to that. It was a really big value proposition for anyone who wanted to access it. Most of the companies went out there and did big TV campaigns, billboard campaigns, or newspaper campaigns; however, the Hotmail team didn't have much funding as they did, so they had to scramble around to figure out how to do it. What they did was add that little link at the bottom of every email that said, 'Sent from Hotmail. Get your free email here.'

The interesting thing was that it meant that the payload was low: You email one person at a time, you're not necessarily going to have a big payload. Maybe you send around one of those spam emails, but I'm not sure I'd click on your link. The frequency is high though, because you're emailing the same people over and over, which means you're going to hit those people once, twice, three times a day and really bring up the impressions. The conversion rate was also really high because people didn't like being tied to their ISP email. So Hotmail ended up being extremely viral because it had high frequency and high conversion rates.

Another example is Paypal. Paypal is interesting because there are two sides to it, the buyer and the seller side. The other thing that is interesting is that its mechanism for viral growth is eBay. So you can use a lot of things for virality that may not look necessarily obviously viral. If you said to a seller that you were going to send them money - I can't think of a higher conversion rate. Frequency was low, and payload was low. But Paypal did this thing where they gave away money when you got your friends to sign up, and that's how they went viral on the consumer side. They didn't have to do that for sellers, because if I said 'I am going to send you money via this,' you will take that. And even on the consumer side they went viral because if someone says 'Sign up for this thing and you'll get ten bucks.' Why wouldn't you? So they were able to go viral because their conversion rate was high on the buyer and the seller side, not because their payload and frequency was high. Make sense?

This is a really good way to look at virality if you want to say, 'Is this product viral?' Facebook was not viral via email sharing or anything like that. Facebook was purely viral via word of mouth. The interesting thing about Paypal and Hotmail, is to use them, the first person has to send an email to a person who wasn't on the service. With Facebook, there is no native way to contact people who aren't on the service. Everyone thinks that Facebook is a viral marketing success, but that's actually not how it grew. It was word of mouth virality because it was an awesome product you wanted to tell your friends about.

Q: In the first round, it makes sense for there to be a low payload. Will the payload increase in later rounds as the campaign grows and people send more and more e-mails?

A: First and foremost, I think you only send emails to a small number of people. So compared to the massive viral engines that exist today, where you import someone's entire contact book and send them all an e-mail, or where you post to everyone's friends on Facebook, the actual payloads are still very small even if it's everyone that you e-mail on a frequent basis you hit. I'm also thinking per email sent out, how many people are on it. But it's a fair point that as more people get on Hotmail, they'll send more emails, and as more people use email, the product grows more and more successfully.

Q: Does a point of conversion matter as well?

A: On Hotmail you click to sign up, but on a billboard you have to remember the URL, go to the website, type it in, find the registration button, click register and sign up. Anything you can do to move friction out of the flow, do it. Going from a billboard ad to an online ad removes huge amounts of friction from the flow.

Q: Are frequency and conversation rate related?

A: Absolutely. If you hit someone with the same email over and over again, or the same banner ad, the same rules apply to every channel. The more times you hit someone with the same Facebook ad, the less they'll click. That's why we have to, like creative exhaustion, rotate creatives on Facebook. Same with banner ads and news feed stories. The fiftieth time you see that IQ story on your news feed, you are not going to want to click on it. The same is true with these emails. So if you send the same email to people over and over again with an invite, you will get a lower conversion rate. 'The more you hit someone with the same message, the less they convert' is fundamental across every online marketing channel.

Second way to look at virality, which I think is awesome, is by this guy Ed. Ed runs the growth team at Uber now; he was at the growth team at Facebook. He was a Stanford MBA student, and did a class similar to this where they talked about virality and built viral products. The interesting thing is, if you look at Uber, they're incredibly focused on drivers. It's a two-sided market place, so they need drivers. It's a huge part of their focus as a team, even though they've got probably the best viral guy in the world at the company.

So with virality, you get someone to contact import (35:12) let's say. Then the question is, how many of those people do you get to send imports? Then, to how many people? Then, how many click? How many sign up? And then how many of those import. So essentially you want people to sign up to your site to import their contacts. You want to then get them to send an invite to all of those contacts - ideally all of those contacts, not just some of them. Then you want a percentage of those to click and sign up. If you multiply all the percentages/numbers in every point in between the steps, this is essentially how you get to the point of 'What is the K factor?' For example, let's says 100 people get an invite per person who imports, then of those, 10% click, and 50% sign up, and of those only 10 to 20% import, you're going to be at 0.5 -

1.0 K factor, and you're not going to be viral. A lot of things like Viddy were very good at pumping up stories. They got the factor over 1, which is perfectly doable. But if you've got something that doesn't have high retention on the backend, it doesn't really matter. You should look at your invite flow and say 'okay, what is my equivalent to import, how many people per import are invites sent to, how many of those receive clicks, how many of those convert to my site, how many of those then import,' in order to get an idea of you K factor. The real important thing is still to think about retention, not so much virality, and only do this after you have a large number of people retained on your product per person who signs up.

A couple more things we are going to touch on: SEO, emails, SMS, and push notifications.

In SEO, there are three things you need to think about. First one is keyword research. People do this badly all the time. So I launched this cocktail site I told you about, I spend a year optimizing it to rank for the word cocktail making, but it turns out in the UK, no one searches for cocktail making- about 500 a month; I dominated that search, it was awesome! 400 visitors a month, it was amazing. Everyone searches for cocktail recipes, and in the U.S., everyone searches for drink recipes. So I optimized for the wrong word. You have to do your research first about what you're going to go after.

Research consists of, what do people search for that's related to your site, how many people search for it, how many other people are ranking for it, and how valuable is it for you? Supply, demand, and value. So, do your keyword research to figure out which keyword you want to rank for. There are many great tools out there. Honestly the best one is still Google AdWords keyword planner tool.

Once you've done that, the next most important thing is links. Page ranks is essentially how all SEO is driven, and Google is based on authority. Now there's a lot of other things in Google's algorithm now, like, do people search for your website, there's a lot of stuff about what the distribution of what the anchor text is that's sent to your site, so that if you abuse it or spam it, they can pop out with spam. White text on a white background five pages below the fold doesn't work anymore.

But the single most important thing is to get valuable links from high authority websites for you to rank in Google. Then you need to distribute that love inside your site by internally linking effectively. We launched SEO in September 2007; I joined Facebook November 2007. When we launched it, but we were getting no traffic from the pages we had launched, public user profiles. So when I went in and looked at it, the only way you could get into any public user profile was to click on the foot of the page for the about link, then click on the blog articles, then click on one of the authors, and then spider out through their friends to get all their friends.

Turns out that Google was like, 'They bury these pages, they're not very valuable. I'm not going to rank them.' We made one change: We added a directory so that Google could quickly get to every page on the site, and we 100Xed SEO traffic. Very simple change drove a lot of upside by distributing the link love internally.

The last thing is that there's a whole bunch of table stakes stuff for XML sitemaps, and making sure you have the right headers; it's all covered really well online for you.

Email is dead for people under 25 in my opinion. Young people don't use email. They use WhatsApp, SMS, SnapChat, Facebook; they don't use email. If you're targeting an older audience, email is still pretty successful. Email still works for distribution, but realistically, email is not great for teenagers - even people at universities. You know how much you use instant messaging apps, and how little you use emails. And you guys are probably on the high scale for email because you're in Silicon Valley. That being said, on email the things to think about: Email, SMS, and Push Notifications all behave the same way. They all have questions of deliverability, so to finish to finish first, first you have to finish. Your email has to get to someone's inbox. So if you send a lot of spam, end up with dirty IPs, or send email from shared servers where other people are sending spam from, you are going to end up being put in the spam folder consistently and your email will fail completely. You may end up being blocked and have your email bounce. There's a lot of stuff around email where you have to look when you receive feedback from the servers you are sending emails to, 500 series errors versus 400 series errors; you have to be respectful how those are handled. If someone gives you a hard bounce, retry once or twice and then stop trying because if you are someone who abuses people's inboxes, the email companies spam folder you, and it's very hard to get out. If you get caught in a spam house link, or anything like that, it's very hard to get out. It's really important with email that you are a high class citizen, and that you do good work with email because you want to have deliverability for the long run.

That counts for push notifications and SMS, too. With SMS, you can go buy SMS traffic via grey routes with people who are having phones strung up attached to a computer and pumping out SMSs. That works for a time, but it always gets shut down. I've seen so many companies make these mistakes where they think they're going to grow by using these kinds of tactics. If you can't get your email, SMS, or Push Notification delivered, you will never get any success from these. You actually spam your power users and give them notifications they don't care about, making it really hard for them to opt out. Well, they start blocking you, and you can never push them once they've opted out of your Push Notifications. And it's very hard to prompt them to turn them on once they've turned them off.

So number one thing to think about regarding email, SMS, and Push Notifications is, you have to get them delivered. Beyond that, it's a question of open rate, click rate. So what is the compelling subject line you can put there so the people can open your email, and how can you get them to click when they visit?

Everyone focuses towards doing marketing emails that are just spam in my opinion. Newsletters are stupid. Don't do newsletters because you'll send the same newsletter to everyone on your site. Someone who signed up to your site yesterday versus someone who's been using your product for three years - do they need the same message? No.

The most effective email you can do is notifications. So what are you sending? What should you be notifying people of? This is a great place where we're in the wrong mindset. As a Facebook user, I don't want Facebook to email me about every 'like' I receive, because I receive a lot of them since I have a lot of Facebook friends. But as a new Facebook user, that first 'like' you receive is a magic moment. Turning on notifications throughout all of our channels, increased on our emails, SMS, and Push Notifications, but we only turned it on for low-engaged users who weren't coming back to the site, so it wouldn't be spamming for them.

So it was a great experience to think about that. The first thing you need to think about when sending emails, SMS, and Push Notifications is what notifications should we be sending. The second thing you need to be thinking about is how can you create great triggered marketing campaigns. When someone created their first cross-border trade transaction was one of the best email campaigns I was ever part of at eBay in terms of click through rate. It was awesome because it was really timely, and really in context - the right thing to do for the user.

I'd say make sure you have deliverability. Focus on notifications and triggered based emails, SMS, and Push Notifications.

There's one thing I wanted to finish with, which is my favorite quote by General Patton. It's so cliche; it's crazy, but it's awesome.

"A good plan, violently executed today, is better than a perfect plan tomorrow."

And one other thing that Chamath instills in us and Mark still instills across the whole of Facebook is move fast and don't be afraid to break stuff. If you can run more experiments than the next guy, if you can be hungry for growth, if you can fight and die for every extra user and you stay up late at night to get those extra users, to run those experiments, to get the data, and do it over and over and over again, you will grow faster.

Mark has said he thinks we won because we wanted it more, and I really believe that. We just worked really hard. It's not like we're crazy smart, or we've all done these crazy things before. We just worked really really hard, and we executed fast. I strongly encourage you to do that. Growth is optional.

Lecture 7: How to Build Products Users Love, Part I

Kevin Hale

Alright, so when I talk about making products that users love, what it means specifically is "How do we make things that have a passionate user base, that our users are unconditionally wanting it to be successful, both on the products that we built and the companies behind them?" We're going to go over tons of information; try not to take too many notes - mostly just try to listen. I'll post a link to the slides on my Twitter account, and on that link, there will be a way for you to annotate the slides. So you can ask me questions, and if we don't get to them, I'll answer them after the talk.

So you guys have been listening to a lot about growth over the last several weeks, and to me, I feel like growth is fairly simple. It's the interaction between two concepts or variables: conversion rate and churn. The gap between those two things pretty much indicates how fast you're going to grow. Most people, especially business-type people, tend to look at this interaction in a very mathematical, calculated sort of way. Today I want to talk about these things at a more human scale because in a startup when you're interacting with your users, you have a fairly intimate interaction in the early stages, and so I think there's a different way of looking at this stuff in terms of how we build our products. We'll look at a lot of different examples of that and how it's executed well.

My philosophy behind a lot of things that I teach in startups is, the best way to get to $1 billion is to focus on the values that help you get that first dollar to acquire that first user. If you get that right, everything else will take care of itself. It's a sort of faith thing.

I came to be a partner at YC by a way of being an alumni. I went to the program of Winter 2006 (it was the second-ever program), and I built a product called Wufoo. Wufoo is an online form builder that helps you create contact forms, online surveys, and simple payment forms. it's basically a database app that looks like it's designed by Fisher-Price. What's interesting though is that because it was fairly easy to use, we had customers from every industry market and vertical you can think of including a majority of the Fortune 500 companies.

I ran the company for five years, and then we were acquired by Survey Monkey in 2011. At the time, we were a very interesting acquisition. We were only a team of 10 people at the time, and while we acquired funding here in Silicon Valley through Y Combinator, we actually ran the company from Florida. We had no office, everyone worked from home, and we were an interesting outlier. So each dot here represents a startup (PowerPoint slide) that exited through IPO or acquisition, we are the outlier to the left. The bottom represents the funding amount that they took, and vertical axis is the valuation of the company at the time. To sum it up the average start up raises about $25 million, and the return for their investors is about 676%. Wufoo, raised about $118,000 total, and our return to our investors was about 29,561%.

So a lot of people are interested in what makes Wufoo a little bit different, or how do we run the company differently. And a lot of it was focused on product. We weren't interested in building software that people just wanted to use, that reminded you that you worked in a cubicle because it was a database app at its core. We wanted a product that people wanted to love, that people wanted to have a relationship with, and we were actually very fanatical about how we approached this idea, to the point where it was almost sort of in a science-y way. So what we said was like, "What's interesting about startups in terms of us wanting to create things that people love, is that love and unconditional feelings, are difficult things for us to do in real life. In startups, we have to do it at scale." So we decided to start off by asking, “How do relationships work in the real world and how can we apply them to the way we run our business and build our product that way?”

We'll go over these two metaphors: acquiring new users as if we are trying to date them, and existing users as if they are a successful marriage.

When it comes to dating, a lot of the things that we uncovered, had to do with first impressions. All of you often talk about your relationships in the origin story. You guys will tell me about your first kiss, how you met, how you proposed. These are the things that we say over and over again, basically the word-of-mouth stories for relationships. There are similar things that we do with companies. Human beings are relationship-manufacturing creatures. We cannot help but create, and anthropomorphize, the things we interact with over and over again. Whether it's the cars we drive, or the clothes we wear, or the tools and softwares we use, we eventually prescribe characteristics to it, a personality, and we expected it to behave a certain way - that's how we sort of interact with it.

First impressions are important for the start of any relationship because it's the one we tell over and over again, right? There’s something special about how we regard that origin story. Let me give you an example. If you're on the first date with somebody, and you’re having a nice dinner, but you catch them picking their nose, you are probably not going to have another date with them. But if you're married to someone for about 20 to 30 years, and you catch them on the BarcaLounger digging for gold, you don't immediately call your lawyer, you know what I mean, and say, "We have a problem here, you have to start drawing up papers for divorce." You shrug your shoulders, and say, "At least he has a heart of gold."

So something about first-time interactions means that the threshold was so much lower in terms of pass fail. So in software and for most products in Internet software that we use, first impressions are pretty obvious and there are things you see a lot of companies pay attention to in terms of what they send their marketing people to work on. My argument for people who are very good at product is that they discover so many other moments and make them memorable: the first email you ever get, what happens when you got your first login, the links, the advertisements, the very first time you interacted with customer support. All of those are opportunities to seduce.

So how do we think about making first moments? We actually took this concept from the Japanese. They actually have two words for how to describe things when you're finished with them, in terms of saying, "is this a quality item?" The two words for quality are atarimae hinshitsu and miryokuteki hinshitsu. The first one means taken for granted quality, which basically means functionality. The last one means enchanting quality. Take for example a pen. Something has miryokuteki if the weight of the pen, the way the ink flows out of it, the way it's viewed by the people reading the hand writing from the pen, is pleasurable both to the user of the pen and the people who experience the byproducts, taking it to the next level. Let's start with some examples.

This is Wufoo's login link, and it has a dinosaur on it, which I think is awesome! But if you hover it, the spec has the added benefit of having a tool tip that doesn't tell you how to log in or what it does, but basically "RARRR!" What we noticed about this in early usability stages, is that this put a smile on people’s faces, like hands down, universally. I think a lot of times when we are assessing products we never think about, "Hey, what is the emotion on the person's face when they interact with this?"

This is Vimeo's launching page; this is actually a couple iterations ago, it's the one that I find to be the most beautiful. It lets you know that when you're starting out on this journey with them, it is going to be something different - they do this all over. If you search for the word "fart," as you scroll up and down, it makes fart noises. There's something different, like this site interacts with you, it's a little bit magical, and it’s a little bit different. It's something that you want to talk about.

You don't always have to do it with design. This is a sign-up form for Cork'd, which used to be a social network for people that love to drink wine. On it, it says, "Email address- it's also your sign in name, and has to be legit. First name – what mom calls you. Last name - what your army buddies call you. Password – something you'll remember but hard to guess. Password confirmation – type it again, think of it as a test." It's literally a poem as you fill it out the form. And this is the kind of thing when you're like, "Oh, I like the people behind this, I’m going to enjoy this experience." Now what does it say, when you fill out a form like this, about what their personality looks like it's going to be? And what's disappointing to me is Yahoo forces every product and service under them to use this exact same login form.

Flickr I thought had one of the best call-to-actions. It was, "Get in there!"

This is for Heroku's signup page. I think this is an older version. What's remarkable about it is that what you start getting a feel for, is like scaling up the backend services, it's as easy as dragging up-and-down different knobs and levers. It looks fairly easy to scale.

This is for a room full of computer science people; I think you'll appreciate that. This is Chocolat, a code editor. And they only have one call to action: when the time limit is up, everything in terms of all the features is all the same, except we changed the fonts to Comic Sans, and what they're basically saying is "Hey, we know who our users are, who our real customers are. They're going to be the people who care about this."

This is Hurl, a website for checking HTTP requests, and sometimes the places where you get errors are opportunities for first moments. If you hit a 404, this is what you get: (Unicorn throwing up a rainbow.)

Often times what we do is we create really beautiful marketing materials, but when you actually need documentation, we sort of skimp out on design features. This is something that happens over and over again. A company that gets this right is MailChimp. What they did was they redesigned all of their help guides so that they looked like magazines covers, and overnight basically readership goes up on all these features, and customer support for these things that help people optimize emails, goes down.

Speaking of documentation, Stripe - what's interesting about an API company, is that there is no UX. The UX is actually just documentation, and there are opportunities even in documentation to sort of enchant and amaze. One of the things that I love about them is that their examples are wonderful, but if you log into the app, one of the things that is a super pain for most people is when you're doing most people's APIs, is grabbing your API credentials and keys. And what shocked us is that it says, "If you are logged into the app, we automatically put your API credentials into the examples, so you only have to copy and paste once, when trying to learn their API."

When Wufoo wanted to launch the third version of our API, we realized, "Okay finally this is good enough that we want people to build on top of it." We were trying to figure out how we launch this out to the world that sort of has our personality behind it, because a lot of people usually do things like a programming API contest that give out iPads and iPhones; it makes you look like everyone else. So at our company, one weird value we have is that our cofounders are big medieval nuts, and we would take everyone out to Medieval Times every single year on the anniversary and founding of the company. So we said we have to do something in that flavor. We contacted the guys at armor.com, and said, "Can you forge us a custom battle axe?" We said, if you win our programming contest, you would win one. The result is, people wanted to talk about this. People wanted to say they were working on this because they wanted to say, "I am programming for a weapon." What's cool is we had over 25 different applications created for us, of quality and quantity that we could not have paid for on the budget and time that we had. We got things like an iPhone app, an Android app, a Wordpress Plugin, and all we did was change the way people talked about our origin story of how they interacted with one of our services.

I'm going to shorthand this by saying you should just subscribe to Little Big Details. It's basically tons of screenshots of software that just shows that they are doing it right and being conscientious of the user and the customers.

When it comes to long-term relationships, or marriages, the only research that we ended up having to read is the stuff done by John Gottman. He's been featured in "This American Life," Malcolm Gladwell's books, etc. He's a marriage researcher up in Seattle, and he has an interesting parlor trick that he can do. He can watch a video tape of a couple fight over some issue for 15 minutes, and predict within 85% accuracy rate, whether that couple will be together or not, or divorced, in four years. If he increases that video up to an hour and asks them to also talk about their hopes and dreams, that prediction rating goes up to 94%. They showed these same video tapes to marriage counselors, successfully married couples, sociologist, psychiatrist, priests, etc. They can't predict with random chance, whether people are going to be together or not.

So John Gottman understands something fundamental about how relationships work in the long term, and that basically how we fight even in the short term period can indicate the whole system and what it's going to look like. One of the surprising things he discovered is not that successfully married people don't fight at all; turns out, everybody fights and we all fight about the exact same things: money, kids, sex, time, and others ("Others" are things like jealousy and the in-laws.) To bring this around, you can actually attribute every single one of these to problems to things you see in customer support when you're building out your products, so Money - this costs too much, or I'm having trouble with credit cards. Kids - users' client. Sex - performance, how long you're up and how fast. Others - I said was jealousy or in-laws, so that's competition and partnerships, anything weird happening there, people are going to write to you about. And the reason I like to think about this in terms of customer support is that, in everyone's processing of a conversion funnel, customer support is a thing that happens in between every one of the steps; it's the reason why people don't make it further down there; it's the thing that prevents conversion from happening.

Now as we were thinking through all of these ideas, and as we were building up the company, we realized that there's a big problem with how everyone starts up their company or builds up their engineering teams. There's a broken feedback loop there. People are divorced from the consequences of their actions. This is a result from the natural evolution of how most companies get founded, especially by technical cofounders. Before launch, it is a time of bliss, Nirvana, and opportunity. Nothing that you do is wrong. By your hand, which you feel is like God, every line you write and every code you write feels perfect; it's genius to you. The thing that happens is after launch, reality sets in, and all these other tasks come in to play; things that we have to deal with. Now what technical cofounders want to do is get back to that initial state, so what we often see is the company starts siloing off these other things that makes a startup company real, and have other people do them. In our minds these other tasks are inferior, and we have other people in the company do them.

So for us, what we're trying to figure out is how we change software development so that we inject some values that we don't talk about enough, like responsibility, accountability, humility, and modesty. We call this SDD (Support Driven Development). It's a way of creating high-quality software, but it's super simple; you don't need a bunch of Post-it notes. All you have to do is make everyone do customer support. What you end up having is you fix the feedback. The people who built the software are the ones supporting it, and you get all these nice benefits as a result.

One of them is support responsible developers and designers. When people built the stuff, they give the very best support. Now we are not the first people to think of this. Paul English was a big supporter of this in Kayak. What he did was install a red customer support phone line in the middle of the engineering floor, and it would just ring with customer support calls. People would often ask him "Why would you pay engineers $120,000 or more to do something that you can pay other people a fraction of." He said, "Well, after the second or third time that the phone rings, and the engineer get the same problem, they stop what they're doing, they fix the bug, and they stop getting phone calls about it." It's a way of having QA in a sort of nice, elegant solution.

Now, John Gottman talks about the reason that we often break up with one another is due to four major causes. They are warning signs. He calls them the Four Horsemen: criticism, contempt, defensiveness, stonewalling. Criticism is basically people starting to focus, not just on the specific issue at hand, but on the over arching issues like "You never listen to users" or "You never think about us" all the time. Contempt is when somebody is purposely trying to insult another person. Defensiveness is not trying to take accountability, or trying to make excuses for their actions. Stonewalling is basically shutting down. Stonewalling according to John Gottman, is one of the worst things we can do in a relationship. Often times we don't worry about these things in customer support, criticism or contempt. Defensiveness, you see this all the time especially in companies as they get older. But stonewalling, this is something I see happening with startups all the time. You get a bunch of customer support calls coming in, and you just think, "I don't need to answer, I don't need to respond." That act of not even getting back to them is one of the worst things you can do, and it's probably some of the biggest causes of churn in the early stages of startups.

This is how support worked out with Wufoo. When we were acquired we had about 500,000 users on the system, 5 million people used Wufoo forms and reports whether they knew it or not, and all those people got support from the same 10 people, and usually there was one person dedicated to support a day, for any shift. Resulting in about 400 issues a week, that's about 800 emails. But our response time from 9 AM to 9 PM was between 7 to 12 minutes, from 9 PM to midnight was an hour, and then on the weekend it would be no longer than 24 hours. We carried this up all the way up to the scale.

What a lot of people talk about and often forget about Airbnb, is how they did this interesting thing where they went up to New York, and offered professional photography, and the founders would go up there and actually take pictures of the people's apartments to help them sell more, focusing on the stories about conversion. What most people don't realize is, a lot of times when I saw Joe in the early times of Airbnb, he had a phone headset stuck to his head all the time because he was doing phone support nonstop.

Churn is the story we don't like to talk about. Airbnb's growth really started picking up when they figured out how to match capacity to the demand or the phone calls they were getting into their support system.

At Wufoo we actually constantly did experiments around support because we were so obsessed with it. One experiment we did was, we heard someone here do a talk about how there's a disconnect between the emotions that we have when we need help, and the content and the reaction we get from people when we get help to people, especially online, because they just don't see those nonverbal cues. So she said, unless there's face recognition on the web, we are just always going to be disconnected from our users. Our feeling was like, "We're not face recognition experts, but we think there's another way of getting empathy." So, as form builders we added a drop-down, and what we said was, "what's your emotional state." Our hypothesis was nobody's going to fill this out; we thought this was going to be a lame experiment, but we'll see how it goes. It turned out that this field was filled out 75.8% of the time. The Browser Type drop-down field in comparison was filled out 78.1% of the time. So people were basically telling us, "For my technical support issue, how I feel about this problem is just as important as all the technical details you need to figure out in order to debug it."

We didn't prioritize things or triage things by emotion, so for the most part people didn't game the system. One of the interesting byproducts of it was that we noticed that people started being nicer to us. We went back and looked at the data, did some text analysis and realized when it comes to communicating with people over written words like email, there's only three ways in which you show strong emotions: exclamation marks, curse words, AND ALL CAPS. Sure enough, on all three of those metrics, they've gone down in the way people were talking to us in customer support. Once people had a simple outlet for their emotions, it made them a lot more rational, and a made our jobs much more pleasant as a result.

The other byproduct that is awesome is that you actually build better software when you do this – far better software. This is actually backed up by a whole bunch of research. Jared Spool, at User Interface Engineer (one of the biggest players in the space) says that there's a direct correlation to how much time we spend directly exposed to users and how good our designs get. He said it has to come in this specific way. It has to be a direct exposure. It can't be something where someone generates a report or through a graph. You have to be interacting with them in somewhat real time. It has to be a minimum of every six weeks, and it has to be for at least two hours; otherwise your software will get worse over time. Our developers, the people who are with Wufoo, are getting exposed to our users 4 to 8 hours every single week. What it does is that it changes the way you sort of build software.

Jared Spool has another way of talking about how we build products. Imagine that this represents all the knowledge needed to use your app on a spectrum (PowerPoint slide). This is like no knowledge (far-left) and this is all the knowledge needed (far-right). These two lines are pretty much your interaction with users. This is currently where their knowledge point is (PowerPoint slide), and this is the target point where you're trying to get them to. The gap between the two is called the knowledge gap as Spool calls it. And what's interesting about this is there's only two ways to fix this. That gap represents how intuitive your app is. You either get the user to increase their knowledge or decrease the amount of knowledge that's needed to use the application. And often times as engineers or people who build and work on these products we think let's add new features. New features only means let's increase the knowledge gap.

So for us we actually focused on the other direction. What that meant is that we spent 30% of our engineering time on internal tools to help with our customer support. But often times it was spent on helping people help themselves, like frequently asked questions, or tooltips; things like if you just click the help link, instead of taking you to the generic help documentation page, you go to the specific page that's going to be the most appropriate for what you're working on. We redesigned our documentation over and over again, A/B tested it constantly. One iteration of our documentation page reduced customer support by 30% overnight. It's one of those things where overnight, all the people that work on the product immediately had 30% less work to do.

What happens if you have everyone working on customer support constantly? I talked about in the very beginning that growth is a function of conversion and churn. This is Wufoo's growth curve for the first five years (PowerPoint slide). What's interesting is that we paid no money on advertising or marketing; all of it was done by word-of-mouth growth. The interaction between new users and downgrades are this (PowerPoint slide). It's so slight what it takes, that gap, making that work. What a lot of people keep forgetting is that there's almost no difference between an increase in conversion rate, 1% increase, and 1% decrease in churn; they do the exact same thing to your growth.; however, the latter is actually much easier to do, and much cheaper to do. And a lot of times we neglect this until way far along, and we usually have our B team work on these projects and services.

This is actually not one of the graphs we tracked most of the time at Wufoo, it's not even the one I'm proud of. This is the one I'm proud of because even though we had this nice, awesome curve of growth, this is what allowed us to scale, keep the company small, and have an awesome culture. And that required doing a lot of these things to help people do what they need.

So John Gottman noticed that there was a different type of behavior for relationships, and why people divorced. Basically there were these subsets of people who stayed together 10 to 15 years and all of a sudden divorced. None of the other indicators would show that this was going to happen. He was looking through the data and realized, "Oh there's no passion, there's no fire between these people." When it comes to relationships, they kind of follow the second law of thermodynamics: In an enclosed energy system, things tend to run down, so you constantly have to be putting energy and effort back into it. The way a lot of people think showing people that "I care about you" in products and companies is by doing things like creating a blog or making a newsletter. The thing is we look at these rates and basically it was such a small percentage of our active users, most of them had no idea about the awesome things that we were doing for them. So we built a new tool and called it the Wufoo Alert System. It allowed us to timestamp every new feature that we are building for users, and that every time they would login we would look at the difference between their log in time, or last login in time, and the new features that were implemented, and they would have this message show up, "Hey since you've been gone, here's all the awesome stuff that Wufoo did for you." Hands-down, this was the most talked about feature that I heard every time I went out to talk to users. They would say things like, "Dude I love that 'Since you've been gone' thing. Even though I pay the same amount every single month, you guys are doing something for me almost every week. It's totally awesome; it makes me feel like I'm getting maximum value."

The other thing that we did in addition to having everyone support the people that paid their paycheck, is have them say "thank you." And this was in large part due to us injecting humility and modesty into the equation. Every single Friday we would get together, we would write simple handwritten thank you cards to our users. And I know there are tons of people who would not be sort of excited about doing this; it was a ritual that made all the difference in terms of like having a team that was very tightly knit, and working on stuff that they really cared about. They constantly knew what the mission was for, and why we sort of did what we did. These aren't fancy thank you cards; they're just simple handwritten stuff on index cards, we threw in a sticker, and slapped on a dinosaur on the front of it.

What's interesting is we started this practice as a result of the early days of starting Wufoo. Chris, Ryan, and I were talking to try to figure out what we were going to do to show users that we appreciated them around Christmas, and Chris came up with this idea where he said, "Hey guys, a couple years ago my mom made me write thank you letters to all my relatives for my Christmas presents, and I really didn't like to do it, but the following year all my presents were super good... so I think we should try this for our business and see how it goes."

So that first year we wrote handwritten Christmas cards to all of our users that first year. Second year rolls around, and we have too many customers with just the three founders. We were thinking, "We're kind of screwed; we don't know what we're going to do." Well, we read a book called The Ultimate Question and in it, he talks about focusing on your most profitable users; if you just take care of them, things will work out. So we thought, "That will work out, that's scalable." Basically we only wrote to our highest-paying customers. So January rolls around that second year and one of our longtime loyal users writes to us. He basically says, "Hey guys I really loved the Christmas card you guys sent me the first year, and I just wanted you to know, I haven't received my second card yet and I'm just looking forward to it; I know you didn't forget about me. Thanks a lot." So we were like, fuck. The best way to exceed expectations is not to set any in the beginning; we were sort of in this conundrum. What we decided after thinking about it for a while was that we had to stop doing it just one time a year; it needs to be something that happens every sort of week. And even though we'll never catch up to all of our customers, just a practice of doing it will make all the difference.

I talked a lot about lovey-dovey, touchy-feely stuff that I think a lot of engineers don't like to think about too often, so I'll end on hard business data or research. There's an article that was put out by the Harvard Business Review several years ago by Michael Treacy and Fred Wiersema and in it they talk about the discipline of market leaders. They say there's only three ways that you achieve market dominance, and depending on how you want to achieve that market dominance, you have to organize your company in a very specific way: best price, best product, and best overall solution. For best price, you focus on logistics, so Wal-Mart and Amazon. If you want to be the best product out there, you focus on R&D, Apple is usually a quintessential example of that. Best overall solution is about being customer intimate. This is the path that you see all luxury brands follow, as well as the hospitality industry. What I love about this path towards market dominance is that the third one is the only one that everyone can do at any stage of their company. It requires almost no money to get started with it. It usually just requires a little bit of humility and some manners. And as a result, you can achieve the success of any other people in of your market. That's all I got, thank you very much.

Q: So what do you do when you have a product with many different types of users? How do you build one product that all these users love?

A: There is an interesting fine line for that. What I usually tell people is focus on those who are the most passionate, especially in the early stages. Whatever niche it's going to be in, that's where I'm going to focus on completely. I think Ben Silverman from Pinterest started off with design bloggers. Tailor your thing for them, and eventually you'll figure out universal values that will appeal to a lot of other people. So just start one at a time. And a lot of examples that you see up there, a lot of companies make the mistake of just thinking "Oh I'll just make my app funny." Humor is like really difficult to do. When you want to shoot for something witty, you have to get functionality right. So like the Japanese quality. If you don't have atarimae, don't try to do anything witty, because it'll backfire. So hands down our number one focus in Wufoo was to make everything as easy to use as possible; everything else was just polish.

Q: How do we balance being obsessed with working on product with all the other skills that are needed by a company, such as marketing, branding, etc?

A: If you're working on product, you should also always have this flip-side where you're talking to users. For us inside of Wufoo, the way we got people to talk to users was through customer support. They got to see firsthand whether the features worked or not, and it also impacted everyone else in the company because everyone had a customer support shift, so they had a social incentive to make everything work. There should be no point where you are only focused on product. You should always have time where you work on product, and then you see what users say to you - like ongoing virtual feedback. So be careful when you don't have that.

My feeling on marketing and sales, my feeling is marketing and sales is a tax you pay because you haven't made your product remarkable. Word-of-mouth is the easiest kind of growth, and it’s how a lot of the great companies grow. Figure out how to have a story that people want to tell about your product where they are the most interesting one at the dinner table. And then that person is your sales person. That person is your sales force for you.

Q: How do you make a decision on product and communicate that with your engineering team when there are lots of different directions to go?

A: We just looked at customer support, which is really easy because you see what people are having the most amount of trouble with. You cannot help but get feature requests from people. No matter whatever openings you have in your product or app, people will jam feature requests in there, so you're going to know what they want. Your job as a product person and an engineer is to not just do what they say, because that way you'll just be a slave. You have to figure out and solve what they really want, that deep underlying reason. The thing is if everyone wants to have a different way to go, then ultimately someone's going to figure something out. But also, make the smallest version of each little idea, no longer than 1 to 2 weeks to build it, so you can try it out to see what works and what doesn't. It's dangerous to have multiple product directions that require a lot of time to figure out.

Q: Can you relay the story about how the King for a Day thing was good at Wufoo?

A: Yeah, okay so I don't like hackathons. I think they sort of suck in terms of those done inside of companies because you spend like 48 hours working really hard on something that you're really passionate about, and 99% of them never make it out to production. It's super sad. So we came up with an idea called "King for a Day." It worked over the weekend. How it worked is someone randomly in the company got drawn and they got to be the king. The king got to tell everyone else what to do on product. So everything that was bothering them about Wufoo or any other feature that they wanted to have built, they got engineering, marketing, and advertising resources of everyone in the company to make it happen. And of course we worked with them to figure out what we could do in 48 hours. We would do this one to two times a year. It was a huge hit and a boost to morale because what people most loved, was working on things that they felt made a difference, like, I made a difference to the app. So for us, that's one way that we would divide time for product direction. Sometimes, the people that work for you are the people who have the strongest opinions about where the product should go. And that's a good way to democratize it a little bit, by rotating it around.

Q: You said you guys all work from home, which usually seems like a nightmare. How did you make that work?

A: We all work from home, and we all work around the Tampa Bay area. We would allow anyone to work from anywhere but usually as we tried to recruit them and meet our team, they usually decided to come and move here anyway. Remote working is especially tricky. A lot of people like to romanticize it, especially people who are employees, but the thing is an office gives you a lot of benefits and efficiencies that you now have to compensate for when you have remote working. But remote working also has these sort of efficiencies. For example, I don't have to worry about my employees losing two hours of their day to commuting. So the biggest thing we had to do for remote working is to respect people's time. The way we had it set up is we actually had a 4 1/2 day workweek at Wufoo; half-day on Friday was for all the meetings and stuff. We said, no biz dev meetings, no talking with other outside parties. They'd have to be done on Friday, on that half-day; they couldn't be done in the middle of the week. And then also one day of everyone was already dedicated to customer support. So everyone in our company effectively only had three days each week to actually build and work on whatever they were doing. But I actually firmly believe that if you have three solid days, 8 to 10 hours, when you're only working on what you need to build, you can get a ton of shit done. So, what we said was, you have to respect everyone's time during that three day period.

What we came up with was a 15 minute rule. You could have a chat or a phone call with someone, but it could last no longer than 15 minutes. So if you had some complicated issue that you couldn't figure out, at 15 minutes you are to immediately table that item, and have us discuss it on Friday. You'd move on to the next item on your list. I would say 90% of the time, the item never got brought up on Friday, because usually what would happen is people would sleep on it, and then you would magically say "Hey I found a solution!" Or "Hey that's not a big problem whatsoever." Most problems inside a company don't need to be solved in real time or right away. The only things are like when the site is down or when payments aren't working. Everything outside of that is kind of luxury. So focus on your priorities as much as possible, and as a result our 10 person team did far more than many many other companies.

But it takes extra work to make remote working happen. We are an extremely disciplined team, and I would have to say, there are not many YC companies that have been able to replicate what we do. I think there are only two companies in YC that have been able to replicate our discipline style. It takes more work in a very different fashion. And often allows you to be a little bit lazier, in terms of all these things around productivity.

Q: How do we set up accountability for our employees as a manager?

A: We were profitable nine months after launch, so we had profit-sharing, which makes incentives pretty simple and clear. It would be a multiple of whatever sort of bonus pool that we had, and performance measures would be based on how they did in customer support, on their duties there, and what they said they wanted to accomplish. I don't like process and I don't like a lot of tools to help people to be productive, so the only thing that we had to help people manage their projects is a To-do list. It was a simple text file that we shared in a Dropbox account. Each person had their name on it, and you got to see every time someone updated things on the To-do list. What we said was every single night, you wrote everything you did that day, and on Friday we would just go over "This is what you said last week you were going to do; this is what you actually got done. What are the problems at hand?" And it's super simple. It creates this nice written trail for how to handle stuff, and I don't have to worry about managing them. They set the tone for how they want to be assessed. And for people who are excellent at what they do, it works very very well. And when you actually have problems, it's very easy to fire people. I was fortunate enough not to have to fire anyone at Wufoo, but we were able to correct everyone's behavior very very quickly because we just looked at this and evaluated the problem: "Look this is a pattern of behavior. You've been doing your work at last minute, etc. This is evidence that you've provided to us; all we have to do is describe it back to you." And because everyone in the company sees it, there's social pressure that's put into place to help make it all happen.

Q: How do you hire people that can work remotely and work in this fashion?

A: Pretty easily, you have them work on a side project for you. So you contract them out, and have them work remotely as such. Usually the projects I like to have them work on are about one month long so you can get a good sense of how people manage themselves and work on things. That was always the first assessment; we never did things just by interviews.

The other thing we had to screen them for was their ability to do customer support because not every engineer has those empathy skills to handle that stress. So sometimes I would have people write "break-up" letters to me in an interview, giving them 15 minutes to write it. That way you get a good sense of their writing skills because 90% of what you're doing in customer support is telling customers bad news like "we don't support that feature, sorry," or "no that's not going to work," or "that's not going to be available."

Q: Are there any tricks or experiments that didn't work out in your company?

A: Okay I'll talk about one. So one of the things that we did early on to try to motivate ourselves was - like, we understood the idea of crunch mode, and that it's really bad for people. Like if you're doing the subscription business, you need people to last for the long-term, and in video games, a lot of the time they crunch people for a specific time and they have multiple sprints. Most of the time the deadlines can get super exhausting. You might get an increase in productivity, but the recovery that you need for people is always greater than the productivity you gain. And in a company where you need everyone doing customer support, being on their game, and constantly pushing out features, you don't have time for recovery.

So we were thinking that we wanted to build a company vacation into how Wufoo works to reward our users every single year. So we thought, if the vacation is built in for the recovery, we can have one crunch period before the vacation set up and only do customer support that will sort of scale with people. So the way we did the very first crunch mode was that, it was just between the three founders, and we had each of us draw a 10 item To-do list that would be fairly aggressive. The first person to get through seven of their items would win, and the last person to get through seven of their items, would become what we called "Trip Bitch." Trip Bitch meant that you carried the other person's luggage and got people drinks when you're on the company vacation. So we did that, and during that period, everyone was pretty excited about it. The winner also got to choose the next company vacation. But all of a sudden, Ryan had basically poorly estimated the items on his list and realized very quickly, "I'm going to fucking lose," and he just sort of gave up. So crunch mode, turned out to be blah mode for him because he knew he was going to lose and became pretty demoralized. So as a result of that we decided not to do it in that similar fashion anymore. Good idea that we like to talk about, but it was one that we never did again.

Alright guys, thanks a lot! You can email me at kevin@ycombinator.com.

Lecture 8: Doing Things That Don't Scale, PR, and How to Get Started

Walker Williams

Thanks for having me! I'm Stanley, the founder of DoorDash. It's really amazing to be here, because it wasn't actually that long ago when I sat in your seats. I was class of 2014, graduated in CS, as well as my cofounder Andy. For those of you who don't know what DoorDash is, we're building an on-demand delivery network for local cities. I want to start off with this photo that I took a few months ago. This was the night when we just raised our series A. I took this photo as I was walking back to where I lived; I actually lived in Roble at the time on campus. I took this photo because I realized just how ridiculous the combination of things I was holding in my hand was at the time. I was holding my CS247 homework, my tax forms (it was April – so I had to fill out my tax forms), that yellow speeding ticket, and right below that was a $15 million piece of paper I had just signed from Sequoia. And that kind of summarizes just how ridiculous our journey has been, doing this while I was at Stanford, and then transitioning this to an actual startup. I want to share with you that story today.

It all began two years ago in a macaroon store. It was my junior year at Stanford during the fall quarter. At the time, I was really passionate about building technology for small business owners. I sat down with Chloe, the owner of Chantal Guillon, a macaroon store in Palo Alto at the time, just interviewing her, trying to get feedback on this prototype we'd been working on, and also just learning about what her problems were in general. It was during this meeting when Chloe first brought up this problem of delivery. I remember she brought down this really really thick booklet. She showed me pages and pages of delivery orders, and a lot of these orders she had to turn down because there was no way she could have fulfilled them. She had no drivers, and she ended up having to personally deliver all these orders. That was a very interesting moment for us.

Over the course of the next few weeks, we talked to around 150 to 200 small business owners, and when we brought up this idea of delivery, they kept agreeing with us; they would say, "You know, we don't have delivery infrastructure. It's such a huge pain for us. There aren't any good solutions out there." This led us to wonder, delivery is such a common thing, such an obvious thing; why hasn't anyone solved this yet? Like, we must be missing something here right? We thought it was maybe because people had already tried this in the past, but they failed because there wasn't consumer demand for this. We asked ourselves, "How can we test this hypothesis?" We were just a bunch of college kids at the time. We didn't own trucks or delivery infrastructure or anything like that; we couldn't just build a delivery company overnight right? So how could we test this assumption we had?

We decided to create a simple experiment with restaurant delivery. We spent about an afternoon just putting together a quick landing page. When I went on the Internet, I found some PDF menus of restaurants in Palo Alto. We stuck it up there and added a phone number at the bottom, which was actually our personal cell phone number. And that was it. We put up the landing page and called it PaloAltoDelivery.com. This is actually what it looked liked (PowerPoint slide): It was super simple, ugly, and honestly we weren't really expecting anything - we just launched it. What we wanted to see was just would we receive phone calls, and if we got enough phone calls, then maybe this delivery idea was worth pursuing.

So we put it up there; we weren't really expecting anything, and all of a sudden we got a phone call. Someone called! They wanted to order Thai food. And we're like, “This is a real order; we're going to have to do something about it.” So we're in our cars and we're like, "We're not doing anything right now, might as well swing by, pick up some Pad Thai, and let's try to see how this whole delivery thing works." And we did. We delivered it to some guy up on Alpine Road I remember. We asked him, "How did you hear about us, what do you do?" He told us he was a scholar, and then he handed me his business card and told me he was the author of a book called Weed the People. That was our first ever delivery. It was like the best delivery/worst delivery you could ever ask for.

And then yeah, the next day we got two more phone calls. The day after that we got five, then it became seven, and then it became ten. And then soon we began to gain traction on campus through PaloAltoDelivery.com which is pretty crazy, because think about it: This was just a landing page. You had to look up PDF menus to place your orders and then call in. This isn't exactly the most professional-looking site, yet we kept getting phone calls; we kept getting orders. And that's kind of when we knew that we were onto something. We knew we found a need people wanted when people were willing to put up with all of this.

I think another key point to remember is we launched this in about an hour. We didn't have any drivers; we didn't have any algorithms; we didn't have a backend; we didn't spend six months building a fancy dispatch system – we didn't have any of that. We just launched because at the beginning it's all about testing the idea, trying to get this thing off the ground, and figuring out if this was something people even wanted. And it's okay to hack things together at the beginning.

At YC there's a mantra we like to talk about that is doing things that don't scale. So at the beginning we were the delivery drivers. We would go to class, and then after we would go deliver food. We were the customer support; you know I sometimes had to take phone calls during lectures. We spent afternoons just going down University Avenue just passing out flyers trying to promote DoorDash. We didn't have any dispatch system so what we had to do was use Square to charge all of our customers. We used a Google Doc to keep track of our orders. We used Apple's Find My Friends to keep track of where all of our drivers were. You know, just stuff like that, just hacking together solutions to try to get this thing off the ground. In fact at one point we were growing so fast that Square actually shut us down because we were under suspicion for money laundering. I mean think about it, we were getting small chunks of $15-$20 orders coming in at a rapid pace. Luckily, my cofounder Tony worked at Square so he just emailed some buddies there and everything was solved.

Another thing about doing things that don't scale is it also allows you to become an expert in your business, like driving helped us understand how the whole delivery process worked. We used that as an opportunity to talk to our customers, talk to restaurants. We did dispatching which helped us figure out - you know, we manually dispatched our drivers and that helped us figure out what our driver assignment algorithms should look like. We did customer support ourselves, getting real-time feedback from customers. I remember for the first few months when we got started, we would manually email every single new customer at the end of every night asking how their first delivery went, and how they heard about us. We would personalize all these emails: If I saw someone order chicken skewers from Oren's Hummus, we would say "Oh I love Oren's Hummus. How are your chicken skewers? How did you hear about us?" Feedback like that was really valuable, and customers really appreciated that.

I remember this one time - this was during YC - we had just come out of a meeting with one of our restaurant partners, and we heard about this ice cream place that had just opened up on University Avenue called Cream, and we wanted to go try it out. Then all of a sudden, our cofounder back at our office/house texted us saying "Oh we need drivers on the road; we got a huge spike in demand." So we debated for maybe about 10 seconds if we should go get ice cream or should we go deliver. We obviously went to deliver, but that kind of became our motivation on scaling, like you know, if we would scale, then we could go get ice cream next time.

Now of course we scale across different cities. Now we have to worry about building automated solutions, building dispatch systems, and figuring out how to match demand and supply - all that fancy technology stuff. But none of that mattered at the beginning because at the beginning it's all about getting the thing off the ground, and trying to find product-market fit.

Just to summarize, there are three things I would say I learned from doing DoorDash. First, test your hypothesis. You want to treat your startup ideas like experiments. The second thing is, launch fast. We launched in less than an hour with a really simple landing page. And finally, it's okay to do things that don't scale. Doing things that don't scale is one of your biggest competitive advantages when you're starting out, and you can figure out how to scale once you have your demand. And then maybe once you've scaled, then you go get that ice cream. Thank you

Q: How did your first customer hear about you?

A: Our very first one, I have no idea. We just launched in Palo Alto; we didn't do any marketing, so I assume he just must have typed in “Palo Alto delivery” into the web browser. And then after that, we did barely any marketing. I think I sent out one email to my dorm, and that was about it. It was all through word-of-mouth. And that kind of just validates how strong the need we found was when people are just talking about you, and willing to put up with a terrible user experience, terrible design, and stuff like that.

Q: When you started, it seemed so obvious to you, you were wondering why, what the reason was nobody had done this before. What's your answer now looking back?

A: Looking back I think the biggest thing is mobile. Now everyone has one of those in their pocket, and we saw that trend and thought what if you could design a delivery system that was entirely based off mobile, where you didn't have to have any infrastructure, or delivery fleets. Instead of hiring drivers full-time or purchasing vehicles, what if you could tap into more of an on-demand pool of independent contractors, and only send orders to them when they have time. So that's kind of the insight we had; everything was done through mobile.

Q: Did you know you were going to be a startup, or were you just making some money at first?

A: At the time we were all just really passionate about building technology for small business owners, and obviously this delivery thing came out of an experiment with the landing page. It was literally an experiment. We weren't expecting anything, and it just took off, and we just went with it. And logistics was always something we were really passionate about as well, like logistics of transportation – the perfect fusion of how you can help small business owners through delivery.

Q: Did you launch the mobile site first or the website?

A: We started with this landing page right here which took us about an hour to launch.

Q: How does DoorDash stand out amongst a very competitive space?

A: At the beginning consumer demand was never a problem, even up until now. So for us it's just about finding a need and just focusing on serving that demand. At the beginning competition doesn't really matter.

Q: How long did it take you to get incorporated into a company?

A: We launched in January 2013, and then we did YC that very summer. When we decided to take this idea through YC, we incorporated.

Q: Where do you plan to go beyond food delivery?

A: For us when we started DoorDash, it was always about helping small business owners and figuring out how you served this for any local merchant whether you were a macaroon store, restaurant, or furniture shop. That's still our focus; that's our long-term vision. For now we are just focused on restaurant delivery as a way to scale, but ultimately that's where we want to end up in.

Sam Altman: Next is Walker Williams, founder of Teespring. He's been working with YC for about a year and a half, something like that. I almost rejected him, which sounds like a dumb idea, but now they're making hundreds of millions of dollars in revenue, so very luckily I did not. Walker is also going to talk about doing things that don't scale.

Walker: Thank you guys for having me! My name is Walker; I am the CEO/Founder of Teespring. For those of you who don't know what Teespring is, we are an e-commerce platform that allows entrepreneurs to launch products and apparel brands without risk, cost, or compromise. Today the company is about 180 folks and we ship tens of thousands of products each day. I want to talk to you about one of the most fundamental advantages you have as a start up, and that's that you are able to do things that don't scale.

I define things that don't scale as things that are sort of fundamentally unsustainable; they will not last; they will not bring in the millionth user. Where they break, it's usually time but it could be a number of other things. But it's really growth strategies that won't take you to a million users. There are three places I want to focus on today. First one is finding your first users. The second one is turning those users into champions, and the third one is finding your product/market fit.

So finding your first users: The first thing you have to understand is that there's no silver bullet for user acquisition. You know, everybody, and this includes me when we got started, looks for that dream solution, that paperclip campaign that has tremendous ROI, some accelerating partnership that's going to springboard you into the stratosphere, and affiliate agreement; something that solves it for you. But the reality is for the vast majority of companies and in fact for every company that I've had the chance to speak to the CEO of, that's just not possible - those are unicorns. And most of the companies that from the outside look like they've had this dream growth curve, the reality is that those first users were impossibly hard to get. Let me tell you about the story about this ridiculously unsustainable business.

So this is Teespring in 2012 (PowerPoint). When we first launched, the business couldn't have looked worse. It took days of meetings; we had to offer free designs, and days of revisions back-and-forth, we'd have to launch the product ourselves, we'd have to do the social media, all to sell about 50 shirts to a local nonprofit and generate about $1000 of revenue. Anybody looking in would've said, "You guys have to give up, this is a terrible idea." But as time went on, those users started to add up, and I think something you have to understand is when you first launch a company, just by virtue of the fact that it's a new product, you're going to be bad at selling it right? You've got no idea what the pain points of customers really are. You've never sold that before. You don't have any success stories to point to, or testimonials. Those first users are always going to be the hardest.

And so it's your responsibility as a founder to do whatever it takes to bring in your first users. It's going to be different for every company. The common thread that I hear is, founders need to spend personal time and effort, a lot of their personal time and effort to bring those users in themselves. It could mean a number of things - anything from sending 100 emails a day, getting on the phone and just calling as many people as you can, going through a network like Stanford or Y Combinator. Anything you can do to just get that first user. I really equate it to pushing a boulder uphill. And if you think of a smooth hill when you get started, the incline is the steepest and those first inches are the hardest. But over time as you get farther and farther, the incline steadies out, it gets easier, and eventually you reach a point where you're at the top of the hill and the boulder starts to roll on its own.

And so for those first users, you cannot just focus on ROI in the sense of time. Do not expect to spend an hour and return thousands of dollars. Maybe Stanley was one of those unicorns - really incredible story. But for most of us, those first two users are going to take a lot of handholding, a lot of personal love, and that's okay - that's essential for building a company. The one caveat of that is, I don't recommend giving your product for free. And there are plenty of exceptions to this rule, but in general, cutting costs or giving the product away is an unsustainable strategy I wouldn't recommend. You need to make sure that users value your product. And you know, people treat products that are free in a much different way than a paid product, and often times it can give you a false sense of security like, "Oh we're getting all these users; surely we can convert them to paid."

The second aspect is what happens when you get those users? How do you turn those users into champions? A champion is a user who talks about and advocates for your product. Every company with a great growth strategy has users who are champions. The easiest way to turn a user into a champion is to the delight them with an experience they are going to remember, so something that's unusual or out of the ordinary – an exceptional experience.

The easiest way to do this early, and again something that is completely unsustainable - it's not going to scale forever - is to just talk to those users. People will say this all the time, it's sort of the core tenant of Y Combinator, is talk to users. I cannot stress how important it is that you spend a large chunk of your time talking to users. You should do it constantly, every single day, and as long as possible. Today at Teespring, I'm still the catchall email address, so anytime anybody misspells "support" or writes an email address that doesn't exist, I get that email. And so I still do about 10 to 20 customer service tickets every single day; I spend hours each night reading every single tweet, probably a little bit OCD, but that's okay; I read through all the Teespring communities. You're never going to get a better sense for your products than actually listening to real users. Especially in the early days, the product and the feature set you launch with is almost certainly not going to be the feature set that you scale with. So the quicker you talk to users and learn what they actually need, the faster you can get to that point.

There are three ways to talk to your customers. You can run customer service yourself. Up until Teespring was doing about $130-$140,000 a month, my cofounder Evan and I did everything in customer service. This is one where there's going to be an instinct to quickly pass off, and that's because it's painful. Even today when I open our customer service portal, I have an emotional reaction where my stomach sinks because it sucks talking to so many users who have had a terrible experience, and it's painful that something that you love and put so much effort into, to know you got it wrong or somebody didn't treat them right. But it's so important that you go through that and learn what you need to build, and what you need to fix.

The second step is to proactively reach out to current and churn customers. Churn customers are customers who have left. This is one that often falls by the wayside in the pursuit of new customers, but you want to make sure that your customers are having consistent good experiences; you don't want to take your current users for granted. When a user actually leaves your service, you want to reach out and find out why, both because that personal outreach can make the difference between leaving and staying; sometimes people just need to know that you care and it's going to get better. And even if you can't bring them back, there's a chance that you can learn from the mistakes you made that caused them leave, and fix it so you don't churn users out in the future in the same way.

Finally, the one I'm probably most OCD about is social media and communities. You need to know how people are talking about your brand. You need to try to make sure that when somebody does have a bad experience, and they're talking about it, that you make it right. Problems are inevitable: You're not going to have the perfect product; things are going to break; things are going to go wrong. That's not important. What's important is to always make it right, to always go the extra mile and make that customer happy. One detractor who's had a terrible experience in your platform is enough to reverse the progress of 10 champions. That's all it takes, is one to say, "No you shouldn't use those guys for X reasons," to ruin a ton of momentum.

There are examples in the early days where we would mess up massive orders. We'd print out colors slightly wrong; it would be the wrong size, and it would be half of our GMV for that month. We would know we got it wrong, and the customer would be unhappy, and the instinct was to say that it was only a little bit off, not completely wrong, or that it would be fine. But the reality is you just have to bite the bullet and make sure it's right. And the customers who are originally the most frustrated tend to turn into the biggest champions and the longest term users.

The last one I want to talk about is finding product/market fit. What I mean by that is the product you launch with will almost certainly not be the product that takes you to scale. So your job in those early days of a startup is to progress and iterate as fast as possible to reach that product that does have market fit. And as engineers your instinct is building a platform that's beautiful, clean-code, and that scales. You don't want to write a duct tape code that's going to pile on technical debt. But you need to optimize for speed over scalability and clean code. An example of this is in the early days, we had a couple enterprise customers come in, sort of bigger nonprofits, and say "Hey, we really like your service, but you're missing these fundamental things, so we're not going to use it." And we looked at what it would take to build out those features, and we weren't sure if they were going to work out long-term, but we wanted to try it.

My cofounder Evan, who is our CTO and a million times better developer than I am, ran the math and figured out that if we did it the right way, it was going to take about a month to build out these features. A month for a startup – you live in dog years – a month is a year, and that just wasn't going to do. So he actually went out and duplicated the code-base, duplicated the database, and was able to basically build a completely different product so that he didn't have to worry about the existing users to serve these enterprise customers. We gave them the tool, they on-boarded, and generated a lot of revenue. Eventually we learned what features were core, and we integrated them into the core product. But what would've taken a month, we were able to do in three to four days.

A great rule of thumb is to only worry about the next order of magnitude, so when you have your tenth user, you shouldn't be wondering how you are going to serve one million users. You should be worried about how you're going to get to 100. When you're at 100, you should think about 1,000. It's one of those things where necessity is the mother of invention, so when you hit the breaking point (the Twitter Fail Whale is a great example), and in Teespring there were month-stretches where every single night the site would crash - every night. Every single person on the team would go to sleep with their phone on loud, under their pillows, so inevitably when their phones went off, we could quickly restart the server and go back to sleep; this would happen daily. But the reality is that it was worth it, and you'll end up with these huge pain points and all this technical debt and regret, but it's worth it just to get to that end goal and that product fit faster. You will make it work; you will survive. Those bumps are just speed-bumps, and speed is so so important early.

The lesson that I've been learning lately is that you want to do things that don't scale as long as possible. There's not some magical moment; it's not Series A, or it's not when you hit a certain revenue milestone that you stop doing things that don't scale. This is one of your biggest advantages as a company, and the moment you give it up, you're giving your competitors that are smaller and can still do these things, that advantage over you. So as long as humanly possible, as long as it is a net positive, you need to spend time talking to your users, you need to move as fast as possible in development, but don't give it up willingly; it should be ripped from you.

To practice what I preach, I want to give you guys my email address. If you guys have any questions, if you want to learn about Teespring, or if you want to print some T-shirts (fingers-crossed), just shoot me an email. I'd love to help and I'd love to speak to you.

Walker@teespring.com

The last thing is, we've created an official "How to Start a Startup" T-shirt with Sam. All proceeds are going to Watsi.org. I couldn't miss this opportunity to sell, so if you guys want to grab one of the official tees, just go to teespring.com/startup; it's supporting a great cause.

Thank you.

Q: What convinced you to get into the market of T-shirt printing when there is so much competition in this business?

A: I think there are two factors to it. First, I completely agree. From the outside, people have been telling us that this is a silly idea since day one in every order magnitude we reach. People will come and say, "This is a terrible idea. Why are you doing that?" But the reason why we launched Teespring is because we ran into a personal pain point. We identified a need and found that there were no great current solutions. I was a student at Brown trying to create a "Remember the Bar" shirt for a dive bar that got shutdown, and I realized that nothing matched my needs. And so because I knew that I had that pain point, and I knew there was market fit, and I had seen people adopt the product, I knew there was something there. And it was also one of those things where you could sort of feel the wind on your back where people were adopting the product quickly. The pain point was clearly there; it's not a met need. So I would say that most times, great ideas start out by looking like the silly ideas, and then you can feel out whether or not there is a scalable business by how people are adopting it and whether it is possible to bring customers onboard.

Q: Are nonprofits your biggest customer base?

A: No, today our biggest customer base is entrepreneurs who are trying to build brands and businesses. We have a little over 1000 people who make their full-time living on Teespring today via brands they've launched. And the other side is influencers, so YouTube stars, Reddit communities, bloggers who want to add product merchandise as a way to create a brand and monetize that affinity. Those are our two biggest markets. We still work with a lot of nonprofits, and love working with them. They are still part of our business, but just not the majority.

Thank you.

Sam Altman: Now we have Justin Kan. Justin was the founder of Kiko, and the Justin.tv which became Twitch. He is going to talk about Public Relations.

Justin: I've started a lot of start-ups, but I think you've heard a lot of awesome "how I got started" stories, so I'm going to talk about something very specific that people always have questions about, which is press: how do you get it and how does it work? This is kind of like an abridged version of what we talk about at Y Combinator. Hopefully you guys will find it helpful.

When most people get started with entrepreneurship, they think about press and being in the press as something that happens magically. They think about journalists out there, trying to get the best stories - it's like a meritocracy - which is absolutely not the case.

Before you think about press, one of the things you really want to consider is who you want to reach, as well as your actual goal. I know when I got started I wanted to be in the news because I thought that's what you did as an important company. It turns out that if you don't have any goals, you're not going to achieve them - that's true of pretty much everything. And with press, if you aimlessly just want to be covered, it's not going to do anything for your startup. If you don't have an actual business goal, then it's not a good use of time.

So there are many different goals. With Socialcam, which is a spinoff of Justin.tv, our goal was to be known as like video Instagram and be thought of in that context. When it was time to pitch to our Silicon Valley investors and influencers, we really wanted to get covered in tech press and be positioned as this new, hot social app.

With Exec one of my goals was to get customers. Exec was like a local cleaning service, and our goal was to get people in San Francisco to use it. It wasn't useful to get national press because 99% of those people couldn't use it. So we targeted initially local press like SF Chronicle, who would directly talk to people who could potentially use our app. TwitchTV, which is probably what you guys mostly know, is like ESPN for gamers, kind of like a live stream community for gamers. Our goal was to reach the gaming industry. Now it's like 55 million uniques and people in the gaming industry know about it, but when we started nobody really knew that; it wasn't a place to advertise, we were a very small gaming community. Our goal was to get people in the gaming industry, whether they were developers or advertisers, to think about us as an important place where influencers were. So we really targeted industry trades and game dev blogs. Stuff that the industry was reading.

So what's an actual story? I think there's a bunch of different types of stories, but these are the ones you usually see in startups. Product launches are when you launch a different version of your app.

There's fundraising; for whatever reason, the press loves to write about fundraising even though it's not very interesting. So like if you raise a million dollar seed round you can pretty much get that covered.

Milestones are metrics, like if you've achieved one million dollars a week in revenue. The company that bought Exec just announced that they achieved one million dollars a week in revenue, and it was covered pretty widely. Business stories, which happen when you're already a successful company, The New York Times, The New Yorker, or Business Magazine, will want to cover your startup story. You don't have to worry about that in the beginning.

What I like to call stunts - I don't know if you guys remember, but a couples years ago, this YC company called WePay dropped a block of ice with money frozen in it outside of a PayPal developer's conference because PayPal was in the news for freezing various developers' account. It was widely covered because it was such an interesting thing, you know, it got them in the story, they wouldn't have been talked about in the context of PayPal at all.

Hiring announcements: If you're a big enough company and you hire someone really important, people will want to cover that.

And finally contributor articles which would be you writing some sort of industry overview or some opinion piece, maybe a tech blog, stuff like that.

Basically any of those things can be a story. Something that people usually don't think about when you're trying to start a startup is, when you start a startup, you think that everything you're doing is interesting, but that may not be true for other people. What you really need to think about objectively is, if I wasn't the founder of this company, would I want to read a story about what I'm pitching? So your incremental feature release or your 2.01 feature release might not be interesting just because you added "Find Your Contacts" on Facebook. You really want to take a step back before you invest the time in actually trying to pitch a story, and think, "Will anyone actually really want to read this?" What journalists and bloggers are really looking for are things that people actually want to read.

The other thing is you don't actually have to be very original - your press doesn't have to be original. It just has to be what I like to call "original enough." You don't want to be the second-coolest company to raise $5 million on Kickstarter - the first guy gets all the news. The first video game console to raise $10 million on Kickstarter was huge news because they were the first in that category, even though a lot of people had raised a lot of money for Kickstarter before. Think about your stories in the context of other things that have been written and if they're like novel enough or aren't something that was just written about in the news.

So one of the actual mechanics of getting the story (this is pretty tactical), if you want to get your news in the press, basically there are some really easy steps to follow: Think of getting press like a sales funnel. You're going to talk to a lot of people but not all of them are going to convert – so you shouldn't be upset when one individual person or reporter doesn't write your story. The first thing is you have to think of it like a story. The second thing is you want to get introduced to any reporter or multiple reporters who are going to write about your thing. It's much much easier, just like any business development, to get in touch with them through someone, rather than cold emailing them.

The best thing to do is go to entrepreneurs who were just written about or friends who were just covered on TechCrunch, get them to introduce you to that reporter who wrote about them. The reason that's good is because from the entrepreneur's perspective, the easiest thing in the world to do is to introduce you to the reporter who already wrote about them. They don't need anything else from the reporter, and they're actually doing that person a favor if the story is interesting; it's not like you're asking for intros to investors or people they would want to hire as employees. And then from the reporter's perspective, they're getting introduced to someone who they already vetted as interesting; they're getting an intro from someone who they believed was interesting enough to write about, so by the transitive property, they're going to think you're probably interesting.

So you get an email from this guy who introduces you to the reporter, and you want to get in contact with them with enough time to get them to write a story - let them know probably a week in advance or more, because they're not going to drop everything they're doing to just write about your news. A lot of people, especially first time entrepreneurs, will come and say "Justin I'm launching this product tomorrow. Can you get me in TechCrunch?" That's probably not going to happen unless you already have a relationship. The best thing to do is give yourself some lead time; get that intro in advance.

So once you've set a date for your news to go out – you’re going to launch a product in two weeks - you have this intro, you've set up some sort of meeting, and you really want to get the reporter to invest time and effort into you because there's a sunk cost fallacy at play. Basically, the more time they spend with you, the more likely they are to write about you. The best thing to do is get a face-to-face meeting. Some bloggers don't actually want to meet face-to-face, but like if not that, then get a phone call. The worst thing to do is just have an email exchange because it's very easy for them to forget about it or ignore it.

The next step is actually pitch them. What I do is actually write out the ideal story that I want to see published in bullet points, and memorize it. And when I have a conversation with them if it's in person, the conversation is structured like my outline, and they'll be taking notes. Then they'll go transcribe those notes into a story. So basically what I wrote will be translated into an actual story. By preparing, you can actually control more of the conversation and not forget critical things like mentioning your cofounder's name or the awesome features in your app are.

If I'm doing this over the phone, I'll make sure to have the bullet point list in front of me and I will make sure to walk through a conversation that includes all those things. So you have a pitch, they take notes, and they're going to write the story at this time. The next thing to do is follow-up like a couple days or day before your actual news goes out. You want to send them an email that says "This is the time we're launching the app; thanks for meeting; here's collateral if there are any videos, photos, or screenshot you want to include; how to spell your cofounder's name or your name. I just include all the information that I really care about and bold it. And then that's it. Hopefully the day comes when you press submit on the release to the App Store and at the same time, they released their article on TechCrunch and you are famous.

So a lot of people ask us about PR firms. So I think in the beginning it's kind of like, everything you do in a startup, you want to do yourself. And it's actually pretty easy especially with tech press and bloggers who constantly need new things to write about. I strongly encourage people to try it themselves, and kind of get started by learning the process themselves before they hire anyone. One thing I'll say is that firms can only help you with the contacts and the logistics, but they can't help you know what's interesting about your company, or – I'm never been able to have any one that’s told me what the stories that I'm producing on; they've only been able to give me a list of reporters I might want to contact. So you really have to be responsible about thinking about what's interesting about your company and what you're doing, you know, the roadmap of interesting things that you're working on.

They're also really expensive. I think we were spending about $5,000 and $20,000 a month, and for various firms, it's a lot - for a startup right? It's generally not a good use of money especially in the early days. Getting press is a lot of work, so you should really make sure it's worth it. Getting press is like a vanity metric: It feels like you're being successful because many successful companies like Facebook are covered in the press all the time, but it doesn't actually mean you're successful; it doesn't actually mean you're making money, getting users, or making those users happy.

Sometimes it's a really good strategy for getting your first hundred or two hundred or a thousand customers, but it's really not a user-scalable acquisition strategy, so it's really something that's just a bootstrap; you can't just get like infinity articles written about you. Eventually people are going to get tired of hearing about your company, and usually that happens really quickly right? The pull point about news is that it's new, so it's pretty hard, unless you're like Google, to get covered in the press every week. If you decide it's worth it though, that you do want to have a regular heartbeat of news, you're thinking about what you're doing that matches those seven story types in the future. When I was working primarily on marketing and PR, I would make a schedule on the calendar of when we're going to launch things and make sure to space them out, but have them appear at regular intervals so people didn't want to forget about us and we could maximize our coverage.

You also want to keep your contacts fresh; it's really a relationship business. Once someone writes about you, you should keep up with them because they could potentially write about you in the future. You're more likely to do something for someone that you've already done something for in the past. I would try to establish good relationships with a couple reporters that you could go to for breaking news; it could come in handy later if you're fortunate enough to have people writing negative things about you; your relationships will help you get out your side of the story.

The last thing is kind of like a golden rule or more of like a "pay it forward:" You should help your fellow entrepreneurs get coverage because they'll help you get coverage. The best way to get coverage is really through these introductions. Whenever I'm meeting with reporters, I throw out names of other things I think would be interesting stories for them, and usually that comes back. The reporters like it because you're helping them find interesting stories, and you're more likely to get leads back from those entrepreneurs that you helped out.

If you're interested in learning more about press, here are two resources that I really like: Jason Kincaid, who is a former TechCrunch reporter, goes through a great overview that covers a lot of things I talked about in more depth. And then kind of an evil resource is this book called Trust Me, I'm Lying which was written by a former marketer at American Apparel. He talked about ways that he evilly manipulated the press, but I think it's a pretty good look into the psychology of how stories spread on the Internet; it's valuable to take a look at. Cool, that's basically it.

Q: When is the right time to start worrying about press altogether?

A: The first time I launched my first startup, you know my first products, for a lot of them we got zero attention and we didn't really know how we could get 100 users. I think it's a really fine way to get 100 users, and a lot of companies at YC when they first launch their product, we'll encourage them to get out and do one TechCrunch story to get a few people to see it. It's good to get in the practice. I wouldn't obsess getting coverage in multiple outlets or anything like that in the very beginning.

Q: How much of a role did you guys play in getting the Pokemon thing out?

A: Twitch had this thing called "Twitch Plays Pokemon" where developers set up a Pokémon Gameboy game that was controlled by chat, so millions of people would be typing in A or B, and the character would wander around aimlessly. That was a huge news story. I think what we did was set the stage by having other news stories that someone from the BBC would Google "Twitch" and be like, "What is this crazy thing that everyone on Reddit is talking about?" They would have some context. The other thing is we didn't come up with the idea for Twitch. That was like fortuitous, but we helped give it legs by making the company available to talk to reporters and suggesting follow-up stories, not only about "Twitch Plays Pokémon" with 100,000 people watching, but also stories about when they beat the game or when Twitch played Crystal, the new Pokémon version. We didn't originate the story; it was the community who I think really originated it.

Okay I think that's it, thank you very much!

Lecture 9: How to Raise Money

Marc Andreessen

Sam Altman: I would like to start with a question for Marc and Ron, which is by far the number one question we are going to be looking at today, what made you decide to invest in a founder company?

Ron Conway: Well we have a slide on that-

Sam Altman: Marc can start while we get that up.

Ron Conway: So what makes us invest in a company is based on a whole bunch of characteristics. I have been doing this since 1994, right before Marc got out of the University of Illinois, so SV Angel and its entities have invested in over 700 companies. To invest in 700 companies that means we have physically talked to thousands of entrepreneurs and there is a whole bunch of things that just go through my head when I meet an entrepreneur. I am just going to talk about what some of those are.

Literally while you are talking to me in the first minute I am saying “Is this person a leader?” “Is this person rightful, focused, and obsessed by the product?” I am hoping—because usually the first question I ask is "What inspired you to create this product?"—I’m hoping that it’s based on a personal problem that that founder had and this product is the solution to that personal problem. Then I am looking for communication skills, because if you are going to be a leader and hire a team, assuming your product is successful, you have to be a really good communicator and you have to be a born leader. Now some of that you may have to learn those traits of leadership but you better take charge and be able to be a leader. I’ll switch back to the slide, but let's let Marc.

Marc Andreessen: Yeah, I agree with all of that. There is a lot of detail to this question that we can talk about. And we may be a little different than Ron, well we are different than Ron, in that we invest in across stages. We invest in the seed stage, the venture stage, growth stage. And then we invest in a variety of business models: consumer, enterprise, and a bunch of other variations. There are a bunch of fine grained answers that we could get into if there are specific questions.

Two general concepts I would share: one is the venture capital business is one hundred percent a game of outliers, it is extreme outliers. So the conventional statistics are in the order of four thousand venture fundable companies a year that want to raise venture capital. About two hundred of those will get funded by what is considered a top tier VC. About fifteen of those will, someday, get to a hundred million dollars in revenue. And those fifteen, for that year, will generate something on the order of 97% of the returns for the entire category of venture capital in that year. So venture capital is such an extreme feast or famine business. You are either in one of the fifteen or you’re not. Or you are in one of the two hundred, or you are not. And so the big thing that we're looking for, no matter which sort of particular criteria we talked about, they all have the characteristics that you are looking for the extreme outlier.

The other thing I would highlight that we think about a lot internally, we have this concept, invest in strength versus lack of weakness. And at first that is obvious, but it’s actually fairly subtle. Which is sort of the default way to do venture capital is to check boxes. So really good founder, really good idea, really good products, really good initial customers. Check, check, check, check. Okay this is reasonable, I’ll put money in it. What you find with those sort of checkbox deals, and they get done all the time, but what you find is that they often don’t have something that really makes them really remarkable and special. They don’t have an extreme strength that makes them an outlier.

On the other side of that, the companies that have the really extreme strengths often have serious flaws. So one of the cautionary lessons of venture capital is, if you don’t invest in the bases with serious flaws, you don't invest in most of the big winners. And we can go through example after example after example of that. But that would have ruled out almost all the big winners over time. So what we aspire to do is to invest in the startups that have a really extreme strength. Along an important dimension, that we would be willing to tolerate certain weaknesses.

Ron Conway: Okay, I don't want to over dwell on the slide, but when you first meet an investor, you've got to be able to say in one compelling sentence that you should practice like crazy, what your product does so that the investor that you are talking to can immediately picture the product in their own mind. Probably twenty-five percent of the entrepreneurs I talk to today, still after the first sentence, I don't understand what they do, and as I get older and less patient I say "Backup, I don't even know what you do yet." So try and get that perfect. And then I want to skip to the second column. You have to be decisive, the only way to make progress is to make decisions. Procrastination is the devil in startups. So no matter what you do you got to keep that ship moving. If it's decisions to hire, decisions to fire, you got to make those quickly. All about building a great team. Once you have a great product then it's all about execution and building a great team.

Sam Altman: Parker, could you talk about your seed round and how that went and what you should have done differently for raising money?

Parker Conrad: Sure, actually my seed round, most of the stuff with my current company felt like, from a fundraising perspective, felt like it came together relatively quickly. But actually, one of the experiences I had, I started a company before this that I was at for six years, and my co-founder and I pitched almost every VC firm in Silicon Valley. We literally went to sixty different firms and they all told us no. And we were constantly trying to figure out how should we adjust our pitch? How should we do the slides differently? How do we tweak the story? That sort of thing. At one point there was this key insight that someone gave me when I was pitching at Khosla Ventures and this VC said "Guys," he was looking for some very particular kind of analysis that we did not have on hand, he was like "Guys, you don't get it." He was like, "You know if you guys were the Twitter guys, you guys could come in and just be like blublublubluh and put whatever up here and we would invest in you. But you guys aren't the Twitter guys so you need to make this real easy and have all this stuff ready for us," and all this kind of stuff.

And I took the exactly opposite lesson of what he wanted me to take away from that which was: geez I should really just figure out how to be the Twitter guys and that's the way to do this. So actually one of the reasons I started my current company or one of the things that I found really attractive about Zenefits is, as I was thinking about it, it seemed like a business. I was so frustrated from this experience of having tried for two years to raise money from VCs and I sort of decided, to hell with it. You cannot count on there being capital available to you. This business that I started seemed like one that maybe I could do without raising money at all. There might be a path to kind of, there's enough cash flow it seemed compelling enough that I could do that.

It turns out that those are exactly the kinds of businesses that investors love to invest in and it made it incredibly easy. So I actually think it seems very kind and I said I was an expert for you guys. I don't actually think I am very good at fundraising. It is probably something that I am less good at than other parts of my job. But I think that if you can build the business where everything is moving in the right direction, if you could be like the Twitter guys, like nothing else matters, and if you can't be like the Twitter guys, then it is very hard for anything else to make a difference for things to come together for you.

Ron Conway: Why did that VC say to be like the Twitter guys when the fail whale dominated that site for two years?

Marc Andreessen: Because it worked.

Ron Conway: The other point I want to make is, bootstrap for as long as you can. I met with one of the best founders in tech who's starting a new company and I said to her "Well, when are you going to raise money?" "I might not," and I go, "That is awesome." Never forget the bootstrap.

Marc Andreessen: So I was actually going to close on this, but I'm just going to accelerate it. Parker, I think, gave you the most important thing you will ever hear. Which is also what I was going to say. So the number one piece of advice that I have ever read and that I tell people on these kind of topics is always from the comedian Steve Martin, who I think is an absolute genius, wrote a great book on the start of his career, which obviously was very successful. The book is called "Born standing up", it's a short little book and it describes how he became Steve Martin. And the part of the book is, he says what is the key to success? The key to success is be so good they can't ignore you.

So in a sense, we are going to have this whole conversation and I am sure we will keep having it, but it is beside the point, because if you do as Parker has done and you build a business that is going to be a gigantic success then investors are throwing money at you. And if you come in with a theory and a plan and no data and you are just one of the next thousand, it's going to be far, far harder to raise money. So that is the positive way to put it, is to be so good they can't ignore you. You are almost always better off making your business better than you are making your pitch better.

The other thing, that's the positive way to look at it, the negative way or the cautionary lesson is that, and this gets me in trouble every single time I say it, but I am on a ton of flu medications so I am going to go ahead and just let it rip, raising venture capital is the easiest thing a startup founder is ever going to do. As compared to recruiting engineers, recruiting engineer number twenty. It's far harder than raising venture capital. Selling to large enterprise is harder, getting viral growth going on a consumer business is harder, getting advertising revenue is harder. Almost everything you'll ever do is harder than raising venture capital. So I think Parker is exactly right, if you get in the situation where raising the money is hard, it's probably not hard compared to all the other stuff that is about to follow. It is very important to bear that in mind. It's often said that raising money is not actually a success, it's not actually a milestone for a company and I think that is true. And I think that is the underlying reason, it puts you in a position to do all the other harder things.

Sam Altman: Related to that. What do you guys wish founders did differently when raising money? And specifically, Marc, you mentioned this relationship between money and funding?

Marc Andreeseen: I think the single biggest thing that people are just missing and I think it's all of our faults, we are all not talking about it enough, but I think the single biggest thing entrepreneurs are missing both on fundraising and how they run their companies is the relationship between risk and cash. So the relationship between risk and raising cash, and then the relationship between risk and spending cash. So I have always been a fan of something that Andy Rachleff taught me years ago, which he calls the onion theory of risk. Which basically is, you can think about a startup like on day one, as having every conceivable kind of risk and you can basically make a list of the risks. So you've got founding team risks, are the founders going to be able to work together; then you have product risk, can you build the product; you will have technical risk, maybe you need a machine learning breakthrough or something. Are you going to have something to make it work, or are you going to be able to do that? You will have launch risk, will the launch go well; you will have market acceptance risk, you will have revenue risk. A big risk you get into with a lot of businesses that have a sales force, is that can you actually sell the product for enough money to actually pay for the cost of sales? So you have cost of sales risk. If you are a consumer product, you have viral growth risk. So a startup at the very beginning is just this long list of risks, right, and the way I always think about running a startup is also how I think about raising money. Which is a process of peeling away layers of risk as you go.

So you raise seed money in order to peel away the first two or three risks, the founding team risk, the product risk, maybe the initial watch risk. You raise the A round to peel away the next level of product risk, maybe you peel away some of the recruiting risk because you get your full engineering team built. Maybe you peel away some of your customer risk because you get your first five customers. So basically the way to think about it is, you are peeling away risk as you go, you are peeling away your risk by achieving milestones. And as you achieve milestones, you are both making progress in your business and you are justifying raising more capital. Right?

So you come in and pitch to someone like us. And you say you are raising a B round. And the best way to do that with us is to say I raised a seed round, I achieved these milestones. I eliminated these risks. I raised the A round. I achieved these milestones. I eliminated these risks. Now I am raising a B round. Here are my milestones, here are my risks, and by the time I raise go to raise a C round here is the state I will be in. And then you calibrate the amount of money you raise and spend to the risks that you are pulling out of the business. And I go through all this, in a sense that sounds obvious, but I go through this because it is a systematic way to think about how the money gets raised and deployed. As compared so much of what's happening these days which is "Oh my god, let me raise as much money as I can, let me go build the fancy offices, let me go hire as many people as I can." And just kind of hope for the best.

Ron Conway: I'm going to be tactical. For sure don't ask people to sign an NDA. We rarely get asked any more because most founders have figured out that if you ask someone for a NDA at the front end of the relationship you are basically saying, I don't trust you. So the relationship between investors and founders involves lots of trust. The biggest mistake I see by far is not getting things in writing. You know, my advice on the fundraising process is do it as quickly and efficiently as you possibly can. Don't obsess over it. For some reason, founders get their ego involved in fundraising where it is a personal victory. It is the tiniest step on the way as Marc said. And it's the most fundamental. Hurry up and get it over with.

But in the process, when somebody makes the commitment to you, you get in your car, and you type an email to them that confirms what they just said to you. Because a lot of investors have very short memories and they forget that they were going to finance you, that they were going to finance or they forget what the valuation was, that they were going to find a co-investor. You can get rid of all that controversy just by putting it in writing and when they try and get out of it you just resend the email and say excuse me. And hopefully they have replied to that email anyways so get it in writing. In meetings take notes and follow up on what's important.

Sam Altman: I want to talk a little more about tactics here. Just how does the process go? Can people email you directly or do they need to get an introduction? And how many meetings does it take to make a decision? How do you figure out what the right terms are? When can a founder ask you for a check?

Marc Andreessen: That was six questions. It was a lot of things. Why don't you describe, you will describe seed then I will describe-

Ron Conway: Yeah so, SV Angel invests in seed stage startups, so we like to be the very first investor. We normally invest today at around the million to two million. It used to only be a million. So if we invest two hundred and fifty k, that means there's five or six other investors in that syndicate. SV Angel has now a staff of thirteen people. I do no due diligence anymore, I am not a picker anymore. I just help on major projects for the portfolio companies that are starting to mature. But we have a whole team that processes. We, at SV Angel, end up investing in one company for every thirty that we look at, and we end up investing at about one a week.

I think what's interesting is, we don't really take anything over the transome. Our network is so huge now that we basically just take leads from our own network. We evaluate the opportunity, which means you have to send in a really great short executive summary and if we like that, we actually vote, although I am not in this meeting anymore, but the group actually takes a vote on do we make this phone call. That's how important time is in this process. And if enough of the team at SV thinks it's interesting then they appoint a person to make a phone call to that founder. Usually somebody on our team that has domain experience. If the phone call goes well, bingo! We want to meet you. If SV Angel asks you for a meeting, we are well on our way to investing. If that meeting goes well, we'll do some background checks, back door background checks, get a good feeling about the company, the market that they are going after and then make the commitment to invest. And then start helping get other value-add investors to be part of the syndicate. Because if we are going to have an equal workload we want the other investors in this company to be great angel investors as well.

Marc Andreessen: Okay, so I will talk a little about the venture stage, the Series A stage. I think it's fair to say at this point, the top tier venture capitalists only invest in two kinds of companies at the Series A stage. One is if they have previously raised a seed round. So it's almost always the case when we are doing a Series A investment for the company when the company has a million or two million in seed financing, from Ron and the folks that he likes to work with. Almost always Ron, just to be clear and folks he likes to work with. So if you are going Series A, the first thing you to do is to raise seed, that is generally the way the progression works at this point. Every once in a while we will go straight to a company that hasn't raised a seed round. Really the only times when that happens is when it is a founder who has been a successful founder in the past and is almost certainly somebody we have worked with in the past.

So actually, we have not announced, but we just did one of these we will announce in a few weeks where it was a founder, I was an Angel investor, actually I think Ron was also in the company in 2006. Then the company did it's thing and ultimately was acquired by another big company. And now that team is now starting new things. So in that case we are just going to jump it straight to an A. Because they are so well known and they have a plan all lined up for it. That's the exception, it's almost always preceded by a seed round. The other thing is, I mentioned this already, but we get similar to what Ron said, about two thousand referrals a year through our referral network. A very large percentage of those are referrals through the seed investors. So by far the best way to get the introductions to the A stage venture firms is to work through the seed investors. Or work through something like Y Combinator.

Sam Altman: Speaking of terms. What term should founders care most about? And how should founders negotiate?

Parker Conrad: Probably precisely because of what Marc said, the most important thing at the seed stage is picking the right seed investors because they are going to lay the foundation for future fundraising events. They're going to make the right introductions, and I think there is an enormous difference in the quality of an introduction. So if you can get a really good introduction from an someone that the venture capitalist really trusts and respects, the likelihood that that is going to go well is so much higher than a lukewarm introduction from someone they don't know as well. So the seed stage, probably the best thing you can do is find the right investors and then-

Sam Altman: How does the founder know who the right investors are?

Parker Conrad: Well, I think it's really hard. I think one of the best ways, and not to give a plug to YC, is YC does a very good job at telling you who they think those people are. And can really direct you towards, and I actually found it to be pretty accurate in terms of who you guys have said were going to be the best people, they ended up being the most helpful as we were raising subsequent rounds that provided the best introductions. The people who I thought seemed okay, but were not as highly rated by YC that ended up being the case that they were real duds in the seed round.

Sam Altman: Someday we are going to publish some of these people-

Parker Conrad: Oh gosh there are going to be a lot of upset people if you do.

Sam Altman: So how do you think about negotiation? How do you figure out what the right evaluation of their company should be, what are the terms?

Parker Conrad: Well when I was starting out, I was raising my seed round and I didn't really know. I mean, we had conversations about this. I probably started a little too high on the valuation side. As you guys know, Y Combinator starts this thing called Demo Day. You get all these sort of investors at once who are looking at the company. I started out trying to raise money for a twelve or fifteen million dollar cap. Which is not quite the same thing as a valuation but roughly the equivalent. And everyone thought that was crazy, you know, that's completely nuts. You are too big for your britches, that completely just wouldn't work. So I started working it down a little bit. Within the space of a couple days I decided I was going to raise at nine, and for some reason that hit some magical threshold on the seed stage that it was below ten that it seemed that there was almost infinite demand for the round at a nine million cap. So no one would pay twelve but at a nine million dollar cap it felt like I could have raised ten million dollars. And the round came together in roughly about a week at that point, once I hit that threshold.

There seemed to be, and they fluctuate over time, these thresholds particularly for seed stage companies that investors think that above this level is crazy, it doesn't matter. And there is a rough kind of range that people are willing to pay. You just have to figure out what that is. Just get the money that you need, don't raise any more than you need. And just get it done. At the end of the day, whether you raise a twelve, a nine, or a six, it's not a huge deal for the rest of the company.

Sam Altman: Is there a maximum in the company that you think founders should sell in their seed round, their A round? Beyond what Paul was talking about.

Parker Conrad: I don't know the rules on this stuff. The tricky thing is, it seems like they are particularly rough for a Series A. You are probably going to sell somewhere between twenty to thirty percent of the company. Below, venture capitalist tend to be a lot more ownership focused than price focused. So you might find that it's actually, when companies raise really big rounds it is because the investors says, "Hey listen, I am not going to go below twenty percent ownership but I will pay more for it." Above thirty percent, probably weird things happen with the cap table, like it gets hard, you know, down the line for there to be a firm on the cap table for everyone. Everything seems to come in in that range, so that probably just is what it is. In most cases, in the seed stage from what I have heard, there doesn't seem to be any magic to it, but it seems ten to fifteen percent is what people say, but that is mostly just what I have heard.

Ron Conway: I agree with all of that. I think it is important to get the process over with. But I think it is important for the founder to say to themselves in the beginning, at what point does my ownership start to demotivate me? Because if there is a forty percent dilution in an Angel round, I have actually said to the founder, do you realize you have already doomed yourself? You are going to own less than five percent of this company if you are a normal company. And so these guidelines are important. The ten to fifteen percent is because if you keep giving away more than that there is not enough left for you and the team. You are the ones doing all the work.

Marc Andreessen: We'll actually, we'll walk. We have seen a series of interesting companies in the last five years that, where we just walk, we won't bid simply on the basis that their cap table is already destroyed. Outside investors already own too much. There is a company we really wanted to invest in, but the outside investors already owned eighty percent of it when we talked to them. And it was a relatively young company, they just had done two early rounds that sold too much of the company. Literally, we were worried, and accurately so, it was going to be demotivating for the team to have that structure.

Sam Altman: One more question before we open up for the audience to ask questions. For Ron and for Marc. Could you both tell a story about the most successful investment you ever made and how that came to happen?

Parker Conrad: Other than Zenefits.

Ron Conway: For me clearly, it was the investment in Google in 1999. And we got a googol return out of it. Funny enough, I meet Google through a Stanford professor David Cheriton, who is in the school of engineering and he is still here. He was actually an angel investor in Google and an investor in our fund. Kind of the quid pro quo we have with our investors in the fund is you have to tell us about any interesting companies that you see. We loved it that David Cheriton was an investor in our fund because he had access to the computer science departments deal flow. And we were at this party at Vivek Wadhwa's house in full tuxedo, I hate tuxedos, does anyone here know David Cheriton? Because you know for sure he does not like tuxedos and he was in a tuxedo. But I went up to him and we complained about our attire. Then I said, hey what's happening at Stanford? And he said, well there is this project called backrub, and it's search. It's page search by page rank and relevancy.

Today everyone says pagerank and relevancy is obvious. Back in 1998 that was not obvious, that engineers were designing a product based on this thing called pagerank. All it was was a simple algorithm that said if a lot of people go to that website and other websites direct them there, there must be something good happening on that website. That was the original algorithm. The motivation was relevance. So I said to David, I have to meet these people. He said, you can't meet them until they’re ready. Which was the following May funny enough, I waited, I called them every month for five months. And finally got my audition with Larry and Sergey. Right away they were very strategic. They said, they we'll let you invest if you can get Sequoia, we don't know Sequoia but they are investors in Yahoo and because we are late to market, but we want to know we have a deal with Yahoo. So I earned my way into the investment in Google.

Marc Andreessen: I will tell one on the other side, which is Airbnb, which we were not early investors in, Airbnb is a growth round, we did the first big growth round under Airbnb at about a billion dollar valuation in 2011. And I think that will turn out to be, I believe that will turn out to be one of the spectacular growths of all time. I think this will really be a great company, so I will tell that story because it is not a story of pure genius.

We passed. I don't think we even met with them the first time around, or maybe one of our junior people did. I said earlier that venture capital is entirely a game of outliers. One of the key things of outliers is that their ideas often seem outright nuts up front. So of course having a website where you can have other people stay in your house, if you made a list of the ideas that were the most nuts that would be right there at the top. Well the second most stupid idea you can think of is having a website where you can stay at other people’s houses. Airbnb deeply combines both of those bad ideas. So of course it turns out, they have unlocked an entirely new way to sell real estate, they have unlocked this gigantic global phenomenon. So part was just coming to the fact that we just whiffed on our initial analysis of the idea and the numbers were clearly proving that we were wrong. And the customer behavior was clearly proven that we were wrong.

So one of the philosophies in our firm is we are multistage, a big reason for that is so we can fix our mistakes and we can pay up to get in later when we screw up early on. The other thing I will highlight on, the other reason we pulled the trigger at a high valuation when we did was because we had spent time at that point with the founders, with Bryan, with Joe, and with Nate. And there is a friend of mine who has a great line, he says when people progress in their careers they get bigger and bigger jobs, and at some point they get the really big job. Some of the people grow into the job, and half the people swell into it. And you can kind of tell the difference. There is a point when people just lose their minds. One of the issues with these companies that are super successful and hyper growth companies, Airbnb was sort of the classic case with these super young founders that haven’t ran anything before. How are they going to be at running this giant global operation? We were just tremendously impressed and are today every time we deal with all three of those guys; how mature they are, how much that are progressing. It's like they get more and more mature, they get better and better judgement, and they get more and more humble as they grow. So that made us feel really good, that not just was this business going to grow, but that these were guys who were going to be able to build something and be able to run it in a really good way.

Ron Conway: You know, people always ask me, why do you think Airbnb is going to be such a great company? Its funny, we are obsessing over Airbnb. It’s because all three founders are as good as the other founder. That is very rare. In the case of Google, two founders, one of them is a little better than the other... hey, he is the CEO. Every company has a CEO. Why am I saying this? When you start a company, you have to go find somebody as good or better than you to be the co-founder. If you do that, your chance of success grow astronomically. And that is why Airbnb became so successful, so quickly. The anomaly is Mark Zuckerberg at Facebook. Yes he has an awesome team, but the Mark Zuckerberg phenomenon where is it one person, is the outlier. So when you start a company, you have got to find phenomenal co-founders.

Sam Altman: Okay, audience members.

Q: So obviously the conventional wisdom of why you raise money is you need it. But the more I get off conventional wisdom the more I hear another story on why you raise money. I am actually hearing founders say it is more to facilitate the big exit. Or in the worse case the acqui-hire instead of fizzling out into nothing. To what extent is that accurate thinking?

Ron Conway: Well if you pick good investors who have good rolodexes and good domain expertise in what your company does, they are going to add a lot more value than the money. Those are the types of investors you should be looking for.

Marc Andreessen: So the answer to the question is yes but in a sense it doesn't matter. Because you can not plan these things according to the downside. I mean that is the scenario that you are obviously not hoping for. While the answer is yes, that should probably not play into the decision making process too much, it might enter into which investor to raise money from, it probably doesn't enter into the whether to raise money question that much.

Q: If you intend to start a business that is capital equipment intensive, do you guys have any advice on how to deal with demotivation? So not everything starts in software, viral, or anything else? What should founders do for capital equipment intensive companies?

Marc Andreessen: I would double down on my previous comments on the onion theory of risk and the staging of risk and cash. Which is the more capital has the business, the more intense you have to be about exactly what is going to be required to make a business work and what the staging of milestones and risks are. In that case you want to line up, you want to be very precise on lining up, because the risk is so high that it will all go sideways, right? You want to be very precise on what you can accomplish with your A round and what is going to be a successful execution of your A round. If you raise too much money in your A round that will seriously screw you up, right, later on down the road. Because you are going to raise a C seed then the accumulative dilution will get to be too much. So you have to be precise on every single round, you have to raise as close to the exact amount of money as possible. Then you have to be as pure and clean and precise with the investors as you can possibly be about the risks and the milestones.

This, by the way, is a big thing. I am really glad you asked the question. It kind of goes back to what Parker said. If you walk into our firm, and you have Twitter or Pinterest or you have something, and it's just viral growth and it's just on fire and it's just going to go, those are the easy ones. It's like, let's just put money in it. Let's just feed the beast. But if you walk in and you’re like, I got this really great idea but it's going to take three hundred million dollars staged out over the next five years probably across five rounds. It has a potential of very big outcome, but this is not Twitter. We will still do those but the operational excellence on the part of the team matters a lot more. One of the ways you convey the operational excellence is in the quality of the plan. Back to the Steve Martin thing, be so good that they can't ignore you. The plan should be precise.

Ron Conway: If you are capital equipment intensive, there are ways of borrowing money, in addition to venture capital.

Marc Andreessen: You can kick in venture debt and then later on lease financing, but then again that underlies the need for operational excellence. Because if you are going to raise debt then you really need to be precise on how you are running the company because very easy to trip the convenance on a loan and it'a very easy to lose the company. So it's a thread the needle process. The demands are just a more advanced level of management than the next Snapchat.

Q: What is a sign you should not work with an investor?

Ron Conway: Well it is the inverse of what I said about a good investor. If it is an investor that has no domain expertise in your company, does not have a rolodex where they can help you with introductions both for business development and in helping you do the intros for Series A, you should not take that person’s money. Especially if they are in it just to make money. And you can sus those people out pretty quickly.

Marc Andreessen: I'm glad you asked this question because it brings up a larger point. If your company is successful, at least the ones we want to invest in want to build big franchise companies, so we are talking about a ten or fifteen or twenty year journey. Ten, fifteen, or twenty years you may notice is longer than the average American marriage. This is significant. The choice of key investors, of particular investors who are going to be on the board for a company, I think is just as important as who you get married to, which is extremely important. These are people you are going to be living with, partnering with, relying on, and dealing with in positions, in conditions of great stress and anxiety for a long period of time.

This is the big argument I always make, and I make it all the time, sometimes people believe it, sometimes they don't. If everything goes great, it doesn't matter who your investors are. But almost never does everything always go great. Even the big successful companies, even Facebook, all these big companies that are now considered to be very successful, all along the way all kinds of shit hit the fan over, and over, and over, and over again. There are any number of stressful board meetings and discussions, and late-night meetings with the future of the company at stake where everyone really needs to be on the same team and have the same goals, be pulling at the same direction, have a shared understanding, have the right kind of ethics, and the right kind of staying power to actually weather the storms that come up.

And one of the things you will find is a big difference for first time founders versus second time founders is almost always the second time founders take that point much more seriously after they have been through it once. So it really, really, really matters who your partner is. It really is like getting married, and it really is worth putting the same amount, maybe not the quite as much time and effort as picking your spouse, but it is worth spending significant time really understanding who you are about to be partnered with. That is way more important than did I get another five million in the valuation order, or another two million dollars in the check.

Ron Conway: I know at SV Angel, our attitude is if you invest in an entrepreneur, we are investing for life. Because if we made the right decision, we are going to invest in every company they start. Once an entrepreneur, always an entrepreneur. We actually do consider it a marriage.

Parker Conrad: I always look for, in that first meeting, do you feel like you respect this person and do you feel like you have a lot to learn from this person? Because sometimes when you meet with VCs, you feel like they are slow on the uptake, they just don’t get it. Sometimes you walk in and they have just this incredible amount of insight into your business that you walk out of their like, man even if these guys didn’t invest that sort of hour that I spent with them was such a great use of my time, I felt like I came out with a much clearer picture of what I need to do and where I need to go. And that is such a great microcosm of what the next couple years are going to be like. If you feel like you would really want this person to be really involved in the company, even if they didn't have a checkbook that they brought with them, that is probably a really good sign. And if not, that is probably a really bad sign.

Q: What is the constraint on how many companies you guys have invested in? Time, money, or lack of good companies?

Ron Conway: SV Angel has kind of got comfortable with one a week. You certainly can’t do more that that, and that is a staff of thirteen. So it's really the number of companies.

Marc Andreessen: Ron if you had twice the number of hours, would you invest in twice the number of companies?

Ron Conway: I would advise against that. I would rather just add more value to the existing companies. SV Angel does have a written conflict policy. But when we do end up with a conflict, it is usually because one company has morphed into another space. We don’t normally invest in companies that have a direct conflict. If we do, we disclose it to the other company, to both companies, and keep in mind at our stage, we don’t know the company's product strategy anyway. We probably don’t know enough to disclose, but our conflict policy also talks about this really important word which is trust. In other words, we are off to a bad start if we don’t trust each other. With SV Angel, the relationship between the founder and us is based on trust. If someone doesn’t trust us, they shouldn’t work with us.

Marc Andreessen: Let me go back to the original question. This is the question that is talked about most in our firm. So the main constraint on a top-tier venture capital firm is the concept of opportunity cost. It's the concept that means everything you do, there are a whole bunch of other things that you can’t do. It's not so much the cost of we invest five million dollars into the company, the company goes wrong and we lose the money. That’s not really the loss that we are worried about because the theory is we'll have the winners that will make up for that.

The cost we worry about is, every investment we make has two implications for how we run the firm. Number one rules on conflicts. So our policy for sure on venture and growth rounds is that we don't invest in conflicting companies. We only invest in one company in a category. So if we invest in MySpace, and Facebook comes along a year later, we are out. We can't do it. Every investment we make locks us out of a category. And the nature, that is a very complicated topic when we’re discussing these things internally. You only know the companies that already exist, you don't know the companies that haven’t been founded yet. And god help you if you invested in an early company that was not going to be the winner and you were locked out by the time the winner came along.

The other issue is opportunity cost on the time and bandwidth of the general partners. Going back to the concept of adding value, we are a general firm, we have official partners. A partner can maybe be on ten to twelve boards in total if they are completely, fully loaded. Basically you want to think of it as a ticket that you have a limited number of holes you can punch, every time you make an investment you punch a hole. When you are out of holes to punch, you are done, you can't make any new investments. That is very much how venture capital operates. A way to look at it is, every board slot that our GP’s have at any given time is an asset to the firm. They can be deployed against an opportunity. But every time we make an investment, it takes the number of slots we can punch down by one. So it reduces the ability for the firm to do new deals. Every investment we make forecloses not just the competitive set but other deals that we will simply run out of time. This goes back to what I said earlier. This company is fairly good, it seems fairly obvious that it's going to get venture funding, why didn't you fund it? Well, on its own, if we had unlimited capacity, we probably would have, but relative to getting blocked by a competitive set, and not having that board seat for an even better opportunity, we pass on that basis a lot.

Q: What would convince you to invest in a company with no product to show. What do you make that judgement based on?

Ron Conway: What would convince us, is what usually convinces us, is the founder and their team themselves. So we invest in people first, not necessarily the product idea. The product idea tends to morph a lot. So we will invest in the team first. If it’s pre users, the valuation is going to tend to be correspondingly lower unless one of the founders has a success track record.

Marc Andreessen: For us, if there’s nothing at the time of investment it's always almost, other than a plan, it's always, usually a founder we have worked with before. Or a founder that is very well known. You know, in these conversations, the default assumption is that we're all starting consumer web companies or consumer mobile companies. You know there are other company categories. For example, enterprise software companies, SaaS, application companies. It’s much more common that there is no MVP, it’s much more common that it's a cold start. And it is much more common that they build a product in the A round. There is no point in having an MVP, the customer isn’t going to buy an MVP, the customer actually needs a full product when they first start using it. So the company actually needs to raise five or ten million dollars to get the first product built. But in almost all those cases that is going to be a founder that has done it before.

Q: What is the ideal board structure?

Parker Conrad: We are fortunate to have, there’s myself and my co-founder and a partner from Andreessen Horowitz. I think it removes the fear, creates a little more trust. It removes the fear that someone is going to come in and fire you arbitrarily because it's time for a big company CEO. But in most cases if you trust the people you are working with, it shouldn't really be an issue. Things almost never come to a board vote, and by the time they do it's something is deeply broken at that point any way. Most of the power that VCs have is outside of the board structure. There are protective covenants that are built into the financing round, you can't take on debt, you can't sell the company, there are certain things you can't do without them agreeing to it anyway.

It's less of a big deal than people make it out to be. What I found, as a founder, if things are going well at the company, you have unlimited power vis a vis your investors. No matter what the board structure is, no matter the convenants in the round. If you say, this is what I want to do, I think this is what we need to do. Even if it's a good investor, or a bad investor, they are like, let's make it happen. They want to ride this rocketship with you, and when things go badly it doesn't matter what protections you have built into the system for yourself.

Marc Andreessen: When a company gets in dire straights, it doesn't matter what the terms for the prior round are, they all get renegotiated. I have been on boards for twenty years, I have never been in an important board vote that mattered. It's never been a vote, many discussions, many controversies, never a vote. The decision has always been clear by the end. It's almost always unanimous. And so I think the decision is almost always around the intangibles and not around the details.

Sam Altman: Okay thank you guys for coming today!

Lecture 10: Company Culture and Building a Team, Part I

Alfred Lin

I'll set the stage with some slides and a few comments but the main stage is going to be with Brian when he comes up and talks about how he built the AirBnB culture. So, you are here, you have been following the presentations, so you know how to get started. You have built a team, you know how to build your product, it's off the ground, it's growing. People love it, you figured out how to do that. You figured out to create a special one of a kind company with monopoly powers, that's big. And the market you are chasing after is slightly bigger than the paper airplane business, so you are good, right? So now what?

So we are here to submit that actually culture is the thing that is actually going to be very, very important to scale the business as well as your team. And hopefully, after this talk you will be able to know: What is culture? Why does it matter? How do you create your core values? And think about elements that fit together for core values and culture that create a high performance team. Get some best practices for the culture.

What is culture? Anybody want to take a guess at how one should define this? {Simple values in a team?} Yeah, that's good. Did you look that up because you had a computer and an internet connection? These are some definitions that you will find in Webster's dictionary but we are Stanford. This is kind of a trick question. It's a CS class, the questions are never straight forward.

The real question is, what is the company culture going to be. Culture, we can generally talk about society, about groups, about places, or things. Here we're talking about company culture. So how does one define company culture? We can take the previous definition and modify it a little bit. This is a hint of how we may want to define company culture. Every day blank and blank of each member of the team in pursuit of our company blank. Some people have filled these in with different things. The first blank, A, could be assumptions, beliefs, values, now my favorite is core values. The second blank for the B blank, people said behavior, my favorite is action. How do you act? In pursuit of goals, that's kind of weak, in pursuit of big and hairy audacious goals is a little stronger, but a better definition is in the pursuit of mission.

So now that we have that definition, what do we do with that and why does it matter? This is a quote from Gandhi "Your beliefs become your thoughts. Your thoughts become your words. Your words become your actions. Your actions become the habits. Your habits become your values. And your values become your destiny." If you don't have a good culture in the company you can't pursue your destiny.

Why it matters is that it becomes the first principles you sort of go back to when you make decisions. It becomes a way to align people on values that matter to the company. It provides a certain level of stability to fall back on. And it provides level of trust, people sort of trust each other with, but it also gives us a list with which you should be able to figure out what to do and what not to do. And what the more important thing about that is what not to do. Then finally the other thing that is important is it allows you to retain the right employees. There are people in this world that are not going to be a fit for your company, but if you have good strong culture, and the strong core values, you'll know who you want to retain and who you truly do not want to retain. And if you take the first letter of those it happens to help you move faster.

Another reason, you're thinking that's like all mushy stuff, this is actually more scientific stuff. So here are indices from 1994 to 2013, stock market indices of companies in the S&P500, and the Russell 3000, and the Fortune 100 Best Companies to Work For. All these companies out there and they picked out companies that they believed were the best companies to work for. The stock market returns of those companies happen to be 11.8%, which is almost twice that of the other two indices. And so there is real power in companies that treat their employees well, where there's a lot of trust, there is a lot of strong culture.

So how do you create a set of values and define the culture etc? Get asked that a lot. You have to start with the leader of the company and the founder, and ask yourself what are the values that are the most important to you? Of those things, that are most important in the business? Who are the types of people you like working with? And what are their values? And through that you distill together what a set of values are. Think about all the people that you've never liked working with. What values do they have? Think of the opposite of that. Maybe those should be considered values for your company. Finally remember the values have to support your mission and if they don't support your mission, you’re missing something. Then the last final checks are they have to be creditable, they have to be uniquely tied to match your mission. So at Zappos, in terms of uniquely applied to the mission, we were focused on creating a culture that was going to provide great customer service. So the first core value we had, was to deliver wow through service. We are very specific that we wanted to deliver great customer service and it was going to be a wow experience. And then below that we wanted to serve.

I had a paragraph talking about what we mean by that, we wanted to support them, doing the wow through service, and support people such as our employees, our customers, our brand partners, and investors. On terms of the opposite thing, we generally didn’t like working with arrogant people, so one of our core values at Zappos was to be humble. So those are two examples where we create core values in a way that became credible and uniquely tied to our mission.

So you go through this process, you come up with a few core values, these might be some of them, whether it's honesty, integrity, service, teamwork. It might be a list, you might start with three, might end up with a list of ten, you might have a list of thirty. It’s a good start. When Zappos went through this process we asked all the employees at the time what core values they can identify with and they came up with thirty-seven. We whittled that down to about ten. And it took a year to do this, that's a long time and you might want to ask why. Well if you just come up with the word honesty, give me a break, everybody wants the culture to be honest, nobody is going to say I want to be lied to every day.

Service, what do you mean by service? There’s got to be a lot more depth in this than that. And everybody talks about teamwork, but there's a difference in level of teamwork that you see in an intramural sports team vs a baseball team. How do you dive deeper into teamwork? What are the things that don't work for a team? A lot of it has to do with communication, a lot has to do with things that people of study, and you may one go deeper into that. At Zappos we thought about, well there are a lot of smart people in this room. When they're fighting with each other and trying to figure out who's right and who's not, it’s probably not the best use of time. We wanted everyone to build off each other and help each other make any idea better. The result is that the company gets a better idea, not that any individual person is right. So we wanted to make and still, this idea that it's company first, then your department, then your the team, then yourself. And how do you do that?

We are going to go a level deeper in that. There's another great element of high performing teams that I really like. Which is this pyramid that was created by Patrick Lencioni, and he wrote this book, “The Five Dysfunctions of a Team.” And the reason this is interesting is he talks about the breakdowns of the team. A lot of teams break down because they have no trust and even if you had trust, why do you need trust? If you have trust, you can actually have debates and conflict and get to the right answer. If you don't have conflicts and debate, it's the blind leading the blind. How do you actually know you got to the right answer before you commit to something? So people are not actually wanting to commit, they're afraid of committing.

Let’s say you get to the next level when you are actually able to commit. What goes wrong then? It's usually because people are not held accountable to things that they committed to. And if people are not held accountable to the things that they committed to, then they can’t get results. If you think about the company as a black box and results, whether it's financial, whether you produce a great product, or anything like that as the output, one of the major inputs is the culture of the company. Some other best practices, we are actually going to talk though in the Q and A because I think they are going to blend into the conversation, is that you want to incorporate your mission into your values, we talked about that.

Performance, you need to think a bit harder, deeper, longer about your values than you might initially think you need to. One of the things I think a lot of companies don't actually do is, they interview for technical fit or skill fit, a competency in that realm, but they don’t actually interview for the culture fit, whether someone will actually believe in and follow the mission. I think that is a big, big no no. I think you can have the smartest engineer in the world but if they don't believe the mission they are not going to pour their heart and soul into it. And that's one of the things where if you actually start thinking about culture, from the interview process, to performance reviews, to making sure that’s a daily habit, you get a lot further with making a great culture.

One final point made here, culture, just like customer service or fitness, is like motherhood and apple pie. Everybody wants to provide great customer service, every company wants to have great culture. What they fail to do is make it a daily habit. You just can't be fit, if you don't do it as a daily habit. Eventually you get out of shape, then you get fat, and then you say, Oh I have to go on a crash diet to get back into shape. That doesn't quite work, and the same is true with something like culture. So I think we checked all of these off, so we can go into Q and A with Brian.

Brian Chesky: Hello everybody. It’s quiet in here, I'll be honest, now I feel a little less on edge. Nothing worse than a room full of people really, really quiet staring at you, but now I feel better.

Alfred Lin: Well I did it for five or ten minutes, you can do it a little longer. So Brian, could you talk about the process by which you came to understand that culture was important to AirBnB and building a company?

Brian Chesky: Yeah, so I think one of the things we realized is, to just give you, I won’t tell the full story of how Airbnb came to be. Some of you may know it. So here’s the very short version the story, Airbnb wasn't meant to be the company we were trying to start. I quit my job, I was living in LA. One day I drove to San Francisco, became roommates with my friend from college, from the Rhode Island School of Design, Joe Gebbia, and I had one thousand dollars in the bank and the rent was one thousand one hundred and fifty dollars. So that weekend this international design convention was coming to San Francisco, all the hotels were sold out, so we decided to turn the house into a bed and breakfast for the conference. I didn’t have a bed, Joe had three air beds, we pulled them out of the closet and called it The Air Bed and Breakfast. That's how the company started.

I recall the story ten thousand times by the way, some version of that story, and I didn't think I'd ever tell that a second time. I remember growing up, I also went to college, my parents were social workers and never thought about me going to art school. They had worried that maybe I would not get a job after college, which I’m sure most parents worry about. Make sure you promise me you get a job with health insurance, I end up starting Airbnb. I remember her telling me, I guess you never got the job with health insurance.

The reason I say this though is, Airbnb was never meant to be the big idea. It was meant to be the thing to pay the rent so we could think of the big idea. Along the way, by solving our problem it became the big idea. So alongside that, we won’t touch on how we built a product, that's probably another conversation that's being talked about, you have to build a team and a great company. And in the early days, we had three co-founders, Joe, Nate, and myself. I think of one of the reasons we're successful was that I was really lucky. I don't think it was really lucky to pick up the idea Airbnb and I don’t think we were lucky to be successful once we had a team. I think we could have come up with a lot of other ideas and been just as successful. I think I was lucky in that I found two great people that I wanted to start up a business with, people I admired. That almost intimidated me by how smart they were. I think that's what the first thing is, to build a team that is so talented that they kind of, slightly make you uncomfortable to be with them, because you know you are going to have to raise your game to be with them.

And then when we were working together in the early days, this is 2008, the first thing is we were like a family. You think about founders like parents and the company like a child. That child will manifest in many ways, behaviors that parents have in the relationship. If the parents are functional but not working together then the child is, frankly, going to be pretty fucked up. You don’t want that. You want your culture to be awesome. And so Joe, Nate, and I were a total family the beginning, we usually worked eighteen hours a day, seven days a week. I remember when we were at Y Combinator, we worked together, we ate food together, or even went to the gym together. We may as well have gotten jumpsuit, we didn’t go that far. It was like we were a mission, a special force.

We had this amazing shared way of doing things with amazing accountability, and that was the DNA of a company. And then we were thinking, at some point you build the product to phase two, which is building the company that builds the product. So a lot of the talk is about how to build a product, how to get product market fit. Once people do that, now you have to build a company. It doesn't matter how great your original product idea is, if you cannot build a great company then your product will not endure. As we thought about this, we realized we wanted to build a company for the long term.

We wanted a company that would endure. To do that, we started noticing companies have something in common. Companies around for a really long time had a clear mission. A clear sense of values, and they had a shared way of doing something that was unique to them and was really special. And so Joe, Nate, and I, when we were three people, decided to look around at companies. I noticed Apple, Steve Jobs' core value was that he believed people with passion could change the world. He said our products change but our value never has. We learned about Amazon, we learned about Nike, we learned about companies in the early days. You can even use this to talk about nations. Even a nation has core values and a declaration so that the nation may endure longer. We started to realize that we needed to have intention, culture needs to be designed. And that is how we got connected. Because we were funded by Sequoia, Alfred just joined them with Zappos, and I was told Zappos had an amazing culture. We went to Las Vegas and met up with Tony and we learned about it.

Alfred Lin: So what did you learn?

Brian Chesky: Well, you guys are crazy. The thing we learned, and we were three people, was if culture is a way of doing things, there really are two arts. One is behaviors and those can change maybe fifty years from now. There will be rituals and behaviors that change, be different. But there have to be some things that never change. Some principles, some ideas that endure, that make you, you. And I think of core values, integrity, honesty, those aren't core values. Those are values that everyone should have. But there have to be like three, five, six things that are unique to you. And you can probably think about this in your life. What is different about you, that every single other person, if you could only tell them three or four things, you would want them to know about you. And we realized that when Zappos was one hundred employees, they wrote down these ten core values. The thing I learned from Tony is I wish I didn't wait until I was one hundred employees to write down a core value. I was talking to Sam, he thinks we are the only company to write our core values down before we hired anyone.

Alfred Lin: How long did it take you to hire your first employee?

Brian Chesky: So the first employee was our first engineer, I think we looked for him for four or five months. I probably looked through thousands of people and interviewed hundreds of people.

Alfred Lin: By then, when you hired him, when did you write it? Was it on day one or two or was it three?

Brian Chesky: I think we started working on it around the time of Y Combinator, which would've been January 2009. It was probably a process that evolved over the course of six to seven months. We finished Y Combinator in April 2009, hired our first engineer in July something like that. Probably six months. Some people ask why did you spend so much time on hiring your first engineer. I think bringing in your first engineer is like bringing in a DNA chip to the company. This person, if we're successful, there were going to be a thousand people just like him or her in that company, it still wasn't a matter of getting somebody to build the next three features we need the ship for users. There was something much more long-term and much more enduring which was, do I want to work with one hundred thousand more people like this? Now, you want diversity to play, you want diversity of background, age. You don't want diversity of values, you want very homogenous beliefs. That's the one thing that shouldn't be diverse.

Alfred Lin: So what were these values?

Brian Chesky: Six core values, I'll talk about maybe three of them. So the first core value we talk about is champion in mission. And what it really means is that we want to hire people that are here for a mission. We don't want people here because they think we have a great valuation, they like our office design, they need a job, or they think it's hot. We want people to be here for the one thing that will never change, and that's our mission. And just to tell you a quick story about our mission, Airbnb, a lot of people describe it as a way to book a room or book a house and travel around the world. And that's what we do, but that is not why we do it. To answer the question on what our mission is, is to tell you a story I think it describes it.

In 2012, I met a host named Sebastian, we do these new jobs around the world where we do meet ups. Sebastian is probably late fifties in north London. Sebastian looks at me and says, "Brian there is this word you never use on your website." And I say, "What's that word?" And he says, "That word is friendship. I would love to read a story about friendship." I said, "Okay read me a story about friendship." He says, "Six months ago the brunt of riots broke out in front of my home and I was very scared. The next day my mom called me to make sure I was ok, I said yeah mom I'm ok. And she goes, what about the house?" He says, "The house is OK as well." He said, "Here's the interesting thing, from the time the riots broke out to the time my mom called me was a twenty four hour window of time. In the periods between that time, seven of my previous Airbnb guests called me just to make sure I was okay." He said, "Think about that, seven of my own guests called me before my own mother did."

I think that says more about his mother than his guests. But in this summer on a typical night or a peak night, we wouldhave four hundred twenty five thousand people staying in homes and living together and they were coming from a hundred and ninety different countries in the world which is every country except North Korea, Iran, Syria, Cuba. So just hearing that story, at our core that is what we are about. That’s much more than just booking a room or traveling. That what we are about is we want to help bring the world together. We want to do that by giving a sense of belonging anywhere you go. Our mission is to belong anywhere. So five years from now, twenty years from now, maybe we're still selling rooms and homes to meet in or not, but I can guarantee you we're always going be about a sense of belonging and bringing people together. And that's the more enduring idea.

So when we hire people, the first thing to make sure it is that, if that is your mission, you need to champion your mission. You champion the mission by living the mission. Do you believe in it? Do you have stories about it? Do you use the product? Do you believe in the product? I used to ask crazy questions, one of the crazy questions Sam reminds me of, I use to interview people. So I interviewed the first three hundred employees at Airbnb which people think I’m really neurotic and they may also be true. I used to ask them a question, if you had a year left to live would you take this job? I amended it, people who say yes probably don't like their families. So I changed it to ten years. I feel like you should use whatever time you have left to live. Whatever you want to do in those last ten years you should just do. I really want you to think about that, that was enough time for you to do something you really cared about and the answer doesn't have to be this company. I say fine if what you’re meant to do is travel or start a company just do that, don't come here. Go do that.

So there is this old parable about two men laying bricks. Somebody comes up to the first man and says what are you doing? I’m building a wall. He asks the other guy, he says I’m building a Cathedral. There's a job and there is a calling. We want to hire people not only looking for jobs, but a calling. And that's the first value, champion the mission.

I don’t want to take all the time, I will talk about just one more. The second value relates to being frugal and I will tell you a story. By the way, all the founding stories of your company end up becoming the things that you keep talking about to thousands of people, kind of like a child, that these things keep coming back later in life. So Airbnb, I think Marc Andreessen said in the last talk that it was the worst idea that ever worked. I remember people thinking we were crazy, I remember telling people about the idea, I actually told Paul Graham, I said we have this idea it’s called Airbnb. He asked, people are actually doing this? I said, yeah. Question was, what's wrong with them? So I knew the interview wasn't going well and in the interview Paul Graham I think wasn't going to accept us. And we told the story of how we funded the company and here's how it goes. We're introduced, Michael Seibel, who is a partner of Y Combinator, introduced me and Joe to like fifteen investors in the Valley, including some of the ones that have been here and all of them said no to the company. They could have bought ten percent of the company for one hundred and fifty thousand dollars. They all said no, they thought it was crazy. No one would ever stay in someone's home.

So we ended up funding the company with credit cards and you know those binders kids in school put baseball cards in? We had to put them in those, we had to put them somewhere. That’s how many credit cards we had and we were completely in debt. And in the fall of 2008, we provided housing for the Democratic and Republican National Convention. We had this crazy idea because we weren't selling a lot of homes. In the year after we launched, we had a hundred people a day visiting our website and two bookings, which is generally bad. It’s like releasing a song and a year later three people are listing to it a day. It’s probably not going to be a very popular song. But I believed in it. Joe and Nate believed in it.

So we are completely in debt, right, and we get this idea. We are an air bed and breakfast providing housing for the Democratic Republican National Convention. What if we made a breakfast cereal for like the Democratic National Convention? And we came up with this Obama themed cereal. And we called it Obama-O’s, the breakfast of change. Then we came up with a Republican themed cereal for John McCain. We found out he was a Captain in the navy. So we came up with… Captain-McCain’s, a maverick in every bite. We had zero dollars and without any money, we tried calling General Mills they told us to stop calling them or they would get a restraining order so that didn't work. We found a local, though an alumni of RISD, who made a thousand boxes deal for us. We ended up sending them to press and eventually within a week it got on national television, national news. We made forty thousand dollars selling breakfast cereal.

In 2008 we made five thousand dollars from our website and we made forty thousand dollars selling breakfast cereal. I remember my mom asking so are a you cereal company now. And that wasn't that bad part, the bad part was the honest answer which was technically, yes. But the reason I tell that is our second core value is to be a cereal entrepreneur. I'm sorry for the cheesy pun. But be a cereal entrepreneur. We really mean, is that we believe constraints bring out the creativity. When you raise eight hundred million dollars suddenly all that scrappiness, it's easy to lose that scrappiness. It’s easier for people to tell you I just need this fifty thousand dollar contract. I need this, I need that. When people are desperate and not being a little bit frugal, not being creative, or tell me they can't do something, I'll just take a box of cereal and even the suggestion of Obama-O’s is you need to be scrappy and frugal. So again a lot of the founding DNA of your company becomes these values, these principles. Everyone knows that if you don't give a crap you shouldn't be here. And it doesn't mean you have to give a crap it just means you have to to be here. You also have to be creative to be an entrepreneur and super scrappy. And these are some of the values we learned.

Alfred Lin: So you guys should start to think about questions when we open up to questions from the audience but I have a few more questions to ask. So this all sounds nice, stories are great. The people here are a pretty skeptical group. It’s a CS department class, probably left brain focused. This feels like a softie, right brain focus. How does having a strong culture help you make important, tough decisions?

Brian Chesky: Well I think that having it so, here's the thing about culture. There are three things they never tell you about culture. First thing is they never tell you anything about culture. No one talks about culture and no one ever tells the need to have strong culture. So there's tons of articles about building a great product, there's tons of articles on growth and adaption, and a few things about culture. It’s a mystical thing that's soft and fuzzy. That's the first problem. The second problem is it is hard to measure. Things that are hard to measure often get discounted. These are two very hard things. The third thing, the biggest problem, it doesn't pay off in the short term. If you wanted to start up a company and sell it in one year, the one thing I would tell you to do is fuck up the culture. Just hire people quickly. Culture makes you hire really slowly, makes you deliberate about your decisions that in the near term can slow progress. Putting an investment into the company short term.

First thing is the need to be very clear about what's unique to you that you stand for. Once you do that, you need to hire people that believe in that. You need to make sure you hire and fire based on these values. One thing we do is constantly repeat over and over again when we interview, we want to make sure they are world class and fit the culture. The first thing I used to ask people at the end of the interview, I would say if you can hire, this is a functional question, if you could hire anybody in the world, would you hire the person sitting across from you? If our vision is the best in the world, why aren’t we hiring the best in the world? Every single person is meant to hire a person better than the previous people. You are constantly raising the bar. Then we have separate people called core values interviewers who aren’t in the function. So it you are an engineer, the core values interviewers are never engineers because we don't want them to be biased and say well I know how good they are. And they interview just for values, to make sure that people care about the same thing. And we said no to a lot of really great people because we just didn't feel right about them being with us long term. So that's one of the things.

Some other examples of when we have had to make hard decisions: in mid-2011, we were mostly United States and we had this internet clone funded by these guys called the Samwer brothers, anyone ever hear of them? They basically clone internet companies. They recently went public. They take American websites and clone them and then quickly try to sell them back to you. It’s kind of like putting a gun to your head. So they had done this to Groupon, Groupon was the fastest growing site in the world ever. They stopped doing Groupon and started cloning us, and this is back when we had forty employees, we had raised seven million dollars. They cloned us and in thirty days they hired four hundred people. And they wanted to sell the company and if they couldn’t, they were going to destroy us around the world.

The problem with Airbnb is if we are not everywhere around the world, a travel site not being in Europe is like your phone not having email, it doesn't actually work. So we were kind of in trouble. We had this conversation, it was a pragmatic decision of should we acquire them and then there was the values decision. The pragmatic one probably said buy them because you can't risk losing international, so just guarantee you are going to get international. We ended up not buying them. The reason we ended up not buying them was I just didn't like the culture. I didn’t want to bring in those four hundred people. I felt like we were missionaries and they were mercenaries. I didn't feel like they were doing it for the beliefs, I thought they were doing it to make a lot of money very quickly.

I believed in a war, missionaries would outlast and endure mercenaries. I also felt like the best revenge against an internet clone was just to make them run the company long term. You had the baby, now you've got to raise it. So that's what we ended up doing and that was a very controversial decision. A lot of people were telling me you should buy this company, we didn't and I think it worked out

Alfred Lin: What percentage of revenue comes from Europe?

Brian Chesky: More than fifty percent

Alfred Lin: I think it worked out. Okay, anybody have any questions? I can keep going . No one? One statement we had at Zappos is culture and brand were two sides of the same coin. Airbnb has a great culture and a great brand. Do you care to talk about branding, it’s kind of a weak thing, it's a value we don't tend to focus on.

Brian Chesky: I actually just said that to Sam. I think Silicon Valley is not historically really strong or we don't talk about culture and brand very much. They are two sides of the same coin. So culture, like the principles and the beliefs you have inside a company that you want people to be aligned with long term, whatever happens inside the company eventually comes out, you can’t hold it in. And brand is really the promise outside the company that everyone identifies with. So I think having a clear mission and making sure that you know that mission and the mission comes through the company, is probably the best thing you can do for both culture and values.

And then the second thing you know is that your brand, the way people think about you and your company, is often decided by your brand evangelists who are your employees. If you have a weak culture, we often think that companies that hire employees, people that are deeply passionate, create companies that customers are really passion about. Those are companies with strong brands. And so Zappos had a really strong brand. A lot of companies, Google, that cared deeply about culture, they actually have a question, is this person Google? Which is meant to be like the catch all for the digital culture. It was a very strong culture, Google has a unique, there's no such thing as a good or bad culture, it’s either a strong or weak culture. And a good culture for someone else may not be a good culture for you.

So I think brand is incredibly important as well. Brand is really the connection of you with your customers and so if you have an incredibly strong culture, the brand will come through. The final thing to say about brand is a lot of people that talk about their brand talk about what they sell. So if you’re Apple, one way of doing the same, we sell computers and the new screens are larger and faster and they talk about bits and bytes.

I remember Steve Jobs had this really important talk where he says the way to win, this is 1997, isn't to talk about bits and bytes. The way to win is talk about what we value, what our core values are, we believe passionate people can change the world. And that was how they introduced a different campaign. So Apple, before they had this huge renaissance, became the most valuable company in the world. They did the Think Different campaign, which is basically saying this is what we believe in. That if you buy an Apple computer, you're also saying I believe in this too. And there has to be a deeper core belief, and if that doesn’t happen, you’re a utility. And utilities get sold at commodity prices.

Alfred Lin: The question is how do you know how to communicate this company culture or core value to the outside world?

Brian Chesky: So the question is, how do you communicate what Airbnb does in the earlier days. Well we learned a lot because in the early days we communicated like a utility. We actually said Airbnb is a cheap, affordable alternative to hotels. Our tag line was forget hotels, save money with Airbnb. And over time we felt like that was, this was way back in the day, we felt like that was way too limiting, that it undercut the idea. We changed our tag line to travel like a human. Which we haven’t kept, but it was basically meant to say that we believe in a certain kind of world and we really feel like travel is mass produced. You feel isolated, you feel like a stranger, and we want to bring the world back to the place where it feels like a village again, where the service is coming from other people, where you have this feeling you belong, and you are treated like a human.

No matter how successful you are in life, often traveling will remind you that you are not that successful. Go through TSA, stay in a typical hotel, sometimes you'll have some problems. We really wanted to make people feel special. This is some of the stuff we did in the early days, we did a lot of storytelling. I mean I've probably told the story of Airbnb ten thousand times and this is a something that is kind of related to culture. Someone asked me the other day, what’s the job of the CEO? There are lots of things a CEO does, but what you mostly do is articulate the vision. To articulate a vision you have to develop a strategy and hire people that fit the culture. If you do those three things you basically have a company and that company will hopefully be successful.

If you have the right vision, a good strategy, and the people to get there. And so what you end up doing is articulating the vision over and over. Whether it’s recruiting, talking to investors, getting funding, doing PR interviews, speaking in a class room. You are always reinforcing the values. You’re doing it in an email to a customer. You just do it a thousand times, it changes and gets better and better every time. So it kind of evolved.

Very good question. How do we make sure the hosts are reinforcing the culture of Airbnb? So the answer to that is we do a pretty good job, but not yet an amazing job at that. When we first founded Airbnb, I took the Craig Newmark school of thought, Craig is the founder of Craigslist: anyone should be able to use Airbnb. If you want to rent your place, you should be able to. It turns out, many people believe in our values because we talked about them and believed them. But there were people who rented on Airbnb, not because they believed in the values but instead because they realized they could make a lot of money renting their home. And not everyone was a great culture fit. These people actually did cause us a lot of problems, so this was a lesson for me. I didn't think, it didn’t occur to me in the early days that the hosts had to completely fit our values. We met them, we attract people like us. We realized hosts are like partners and they need to believe in the same culture we do. So now we have this program called the Super Host program where they have to demonstrate values to reach this kind of badge which gets the priority customer support and distribution. We are having this important host convention where we bring all the hosts in and will be talking about reinforcing our values. So the answer is, we were really late, but we now do it by reinforcing it every step of the way.

Q: Brian, Airbnb has made some great contributions to the open source community. Do you have any thoughts on how that contributes to your culture and company values?

Brian Chesky: Yeah, I think just in general, it may be related to two things about Airbnb. We tend to be a pretty open culture just in general, we believe in a shared world where people are giving back, contributing to making communities and industries stronger. It’s my philosophy to talk about everything internally, except for things that relate to customer or client privacy, if it doesn’t relate to those two things, then we will talk about it.

As far as open source culture and engineering, we wanted to make sure that we had a really strong identification of the team and so we really felt like a lot of source code shouldn't be you know, we felt like every company needs some kind of moat that protects you from your competition. We thought some technology would be, but we also felt like we wanted to give back from a technology standpoint. We preferred our moat to be that we provide the very best experience in the world when you use Airbnb, that has the biggest network effects. We thought that kind of took precedent over some of the technology that we use, so we decided to try to share some of that out to people. I think it does relate to the values, I never recommended, I never one day recommended we do any of that. We hired engineers that we thought shared our values in culture and they felt that was the right thing to do.

Q: You talk about conventions, what did you do when you had no money and you only got paid for those who used your site. What did you do to scale that up? How did you get users to the site.

Brian Chesky: So this is not about culture, but I will answer it anyway. The best advice I ever got was probably from Paul Graham. Paul Graham said, I remember he had this line, it's better to have a hundred people that love you than to have a million people that just sort of like you. It's literally better to have a hundred people love you. And the reason why, if you have one million customers, one million users and they just kind of don't care about you but they use your app and think you're okay, to get them to care is a really hard thing, I don’t know how to get a million people to all of a sudden care. But I do know if you get a hundred people who love you, those people, if they feel incredibly passionate, they will go out and eacb tell a hundred people. All movements that lead to companies or ideas start with just a hundred people.

So the reason this is so critical, is he gave us another lesson which is that if all you need do is get a hundred people to love you, is do things that don’t scale. So it's hard if you've got a million people, you can't you meet them all. You can meet a hundred people, you can spend time with them. So that's exactly what we did. Joe, Nate, and I would go door to door in New York City, or in Denver where the Democratic National Convention was, literally staying and living with our users. It's a joke that when you buy an iPhone, Steve Jobs does not come sleep on your couch. But I will. That was really critical, living with our users. All we had to do was get in with them and share a passion with our users. You work backwards from a hundred people, even one person. But with our new technology, imagine what would be an amazing experience for just this one person. Walk through the journey from that one person, make it perfect for that one person. Once you make it perfect for one person, it's actually really easy to make a service easy for one person, it’s not that hard. Where everyone gets in trouble is they try to solve both at the same time. So the first thing we did was went door to door, that is one thing, that set up memories.

I will give one more example before I stop talking about this. Right now with Airbnb you can click a button and put in your home and a professional photographer comes to your home and photographs it for free. We have five thousand photographers around the world and we have photographed hundreds of thousands of homes. So it is the probably one of the largest on-demand photography groups around. I believe this started with Joe and I, we were staying with this one host, in New York City and her house is amazing but her photos were terrible. So we asked, why you don’t put up better photos? This is before the life of the great cameras, 2008. She couldn’t figure out how to get photos from her phone onto her computer. She wasn’t a very technically savvy woman. And I just said, we will just take photos for you. Actually, I said, what if you could press a button and somebody would show up at your door to take professional photographs? She said that would be magic.

So the next day, I knock on her door going, I’m here. I photographed her home. I sent a few emails to people saying we give this new magical photography service and if you want, you can press the button for a professional photographer to show up at your home. So they would hit this button and it would send me this alert. We rented a camera in Brooklyn. And in January 2009, walking through the snow, we photographed people’s homes. We did this by hand without any technology. We managed that with spreadsheets. I wasn’t going to bother Nate to design something for photography. Then we started hiring contract photographers. Eventfully we got an intern to manage all the contract photographers. Then we made a full time position to manage the other interns of the contract employees. And at some point, this is before we built anything, at some point there were too many people to manage. There were like hundreds of photographers. Then we built all the tools to do all the photography. But we did it only after we knew what the perfect service was.

Q: One more question, the question is, in this particular situation with Airbnb, a lot of people think it is not necessarily a technology company, but more of a marketing company.

Brian Chesky: Good question. I will answer the question with a story.

Alfred Lin: Let me preface that question with a series of questions. Do you today have propriety technology?

Brian Chesky: Yes.

Alfred Lin: Do you have a moat?

Brian Chesky: Yes Alfred Lin: Do you have network effects?

Brian Chesky: Yes.

Alfred Lin: Do you have pricing power?

Brian Chesky: Yes

Alfred Lin: Do you have a good brand?

Brian Chesky: I think so .

Alfred Lin: Are you a monopoly?

Brian Chesky: I am not going to answer that one

Alfred Lin: Getting back to the question, just forgetting about all of that, companies that have network effects and sort of get off the ground, the fly wheel is going, people just think you are lucky.

Brian Chesky: Let me just answer that one. It's a totally fair question and people have said it so I want to answer it. The guy who owns Sequoia Capital, his name is Doug Leone. One day, I think it was a year, year and half ago, he says, Your job sucks. And I was all, What the hell does that mean? He says, You have the worst job of any CEO in my portfolio. I said, Tell me why.

And this is what he said. "First of all you are a technology company," and I would say that at our heart we are a technology company. "So you have all the challenges of all the other portfolio companies. But beyond that, you are in a hundred and ninety different countries. So you have to figure out how to be international. You have to hire in every country in the world." We are literally in every country except North Korea, Iran, Syria, and Cuba. We are a payments company. We handle billions of dollars of transactions a year, we had to get a business license in the state of California. We have serious fraud and risk. It needs to be locked down like Fort Knox. He said, "That's usually where companies end. But you have to worry about all the other crap. Trust and safety."

We have four hundred and twenty five thousand people staying in other people's individual beds, in their sheets. Think about a woman from Texas staying in the Middle East, or vice versa. Think about the cultural conflicts that could happen and misunderstandings. You have four hundred and twenty-five thousand people a night. It's like being the Mayor of Oakland. So imagine you are the Mayor of Oakland and all the things that happened in Oakland tonight. You have trust and safety.

Now we have regulatory problems. We are in thirty thousand different cities. Every city has different rules, different laws, and many were written in a different century. They were written before you had all the technology. Then you have issues like search and discovery. Google has this thing about being really good at search. Google can give me all these results, but it's clear that there is only one or two right answers for everybody. We have forty thousand homes in Paris. There is no best home in Paris for anyone in this room. So we have to be really, really great at matching people and technology. Another example, Facebook is a digital product. Their product is their website. Our product is these experiences that you have in the real world. We are not just an online product, but also an offline product. So basically the long and the short of it is we have to be world class at technology, world class at design, we have to be world class at branding. We have to convince government we are good for their neighborhood. Because we have to convince people that we are not crazy, this is a real thing, we have to make sure trust and safety is world class. We handle all these payments and handle risk. And I didn't even mention culture. This was not about culture. I really do not see it as a marketing company.

Alfred Lin: Thank you.

Brian Chesky: Thank you guys.

Lecture 11: Company Culture and Building a Team, Part II

Patrick Collison

Sam Altman: Part two of Culture and Team. We have Ben Silbermann, the founder of Pinterest, and John and Patrick Collison, the founders of Stripe. Founders that have obviously, some of the best thinking about culture and building their teams.

There's three areas we are going to cover today. One will be general thoughts on culture as a follow up on the last lecture. And then we are really going to dig in to the findings of these companies and building out the early team. Then how that changes and evolves as these guys have scaled their companies to the hundred plus, I don't even know how many employees you have now but quite a lot, these very large organizations and how they adapt these principles of culture. But to start off, I just want to ask a very open ended question which is, what are the core pieces of culture that you found to be the most important in building out your companies?

Ben Silbermann: Sure. What are the most important parts? For us, we think on a few dimensions. One is who we hire, what those people value. Two is what we do every day. Why do we do it? Three is what we choose to communicate and I think four is how we choose to celebrate. Then the converse of this is what you choose to punish, but in general I think running a company based on what we celebrate is more exciting than what we punish. I think, the four things I think make up the bulk of it for us.

John Collison: One thing, I think Stripe has placed a large emphasis on, more so than other companies, is transparency internally. I think it's something that's been really valuable for Stripe, and a little bit misunderstood. All the things people talk about, like hiring really great people or giving them a huge amount of leverage.

Transparency for us plays into that. We think that if you are aligned on a high-level about what Stripe is doing, if everyone really believes in the mission, that if everyone has really good access to information, and everyone has a good picture of the current state of Stripe, then that gets you a huge amount on the way there in terms of working productively together. And it forgives a lot of the other things that tend to break as you grow a startup. So as we have grown, we started off two people, we're now a hundred seventy people, we've put a lot of thought into the tooling that goes around, transparency. Because with a hundred and seventy people, there's so much information being produced that you can't just consume it all as a fire hose. And so, how we use email, things like that, we can go into it more later. But that's one of the core things that helped us work well.

Patrick Collison: I think culture, to some degree is a resolution to a bandwidth problem. In the sense that, when you start out working on something you are coding all the time, but you can't code all the things that you think the product may need. And so the organization gets larger. Maybe in some idealized world, I don't actually think this is true, but ideally you should be involved in every single decision, in every single type of moment of the company, everything that happens, but obviously you can't. Or maybe you can if you're two people, but you certainly can't if you're five or ten. At that time, it comes very quickly, by a hundred and fifty, it's completely hopeless.

And so culture is the invariant that you want to maintain, as you get specifically involved in fewer and fewer decisions over time. When you think about it that way, maybe intended importance becomes self-evident. Again, the fraction of things you can be involved in directly, diminishing, exponentially, assuming your headcount growth is on a curve that looks like one of the great companies. And yeah, that's super important. It manifests itself in a bunch of different ways. For example, in hiring, maybe the reason the first ten people you hire, the decisions are so important that aren't just hiring those first ten people, you are actually hiring a hundred people because you think each one of those people are going to bring along another ten people with them. And thinking exactly what ninety people that you would like those first ten people to bring on. It's going to be quite consequential for your company but really briefly I think it's about abstraction.

Sam Altman: One thing a lot of speeches in this class have touched on is hiring those first ten employees. If you don't get that right, the company basically never recovers. But no one has talked about how to do that, so what have you looked for when you have hired these initial employees to get the culture of the company right? How have you found them?

Ben Silbermann: I guess this answer is different for every company. I'll say for us it was very inductive. Clearly I looked for people that I wanted to work with, and I thought were talented. I have read all these books on culture, because if I don't know something, the first thing I do is go read about it. Everyone has all these frameworks, so I think one big misconception that someone said once was, people think of culture as architecture when it is a lot more like gardening. You plant some seeds, then you pull out weeds when it's not working, and they sort of expand. When we first hired people, we hired people that were more like ourselves. I often looked for three to four things that I really valued in people. I looked for people who worked hard, had high integrity, low ego. I looked for people who were creative, super curious, which meant they had all these interests.

Some of our first employees are some of the quirkiest you have ever met. They were engineers but they had all these crazy hobbies. Like one guy made his own board game, with his own elaborate set of rules. Another guy was really into magic tricks, and he not only coded a magic trick on an iPhone but he shot the production video in the preview. And I think that quirkiness is a calling card that we find, the people that are excited about many disciplines and extraordinary at once, tend to build really great products and are really great at collaborating.

And the last thing, we really want someone who wants to build something great. And they aren't arrogant about it, they want to take a risk and build something bigger than themselves. And that, in the beginning, is very easy to select for. If you were in our position, we were in this horrible office, nobody got paid. There was no external reason to stay except wanting to build something to join. In fact there was every reason not to. And that's something, looking back, that I really value. Because you always knew people were joining for the purest of reasons and in fact forgoing other job opportunities, market salary, a clean office, good equipment just for the chance to work here. To this day, I think a lot of those traits are seeded and embedded in the folks that we look at now.

John Collison: The first ten hires is really hard, because you're making these first ten hires at a point where no one's heard of this company before. Nobody wants to work with you. You are these two weird people working on this weird idea-

Patrick Collison: Their friends are telling them not to join. For our second employee, he either accepted the offer or was just about to, and his best friends took him out the night before, it was a full on assault, why you should not join this company. Why it is ruining your life basically. And the guy continued to join, actually one of those friends now works at Stripe, but this is what you are up against.

John Collison: And it's hard, no batch of ten people will have as big of an influence on the company as those first ten people. And I think that everyone's impression of recruiting is you open LinkedIn and it's sort of like ordering off the dollar menu, I want that one, that one, that one and now you have some hires. At least for us, it was over a very long time period talking to people we knew, or friends of friends into joining. We didn't have huge networks, we were both still in college by then. So there were really no people that we worked with to draw in. So a lot of those early Stripers were people we heard of from friends.

The other interesting thing they had in common, they were all really early in their career or undervalued in some way. Think about it, if someone is a known spectacular quantity, then they are probably working in a job and very happy about it. So we had to try to find people who were, in the case of our designer that we hired, he was eighteen and in high school and in Sweden at the time. As the case with our CTO, he was in college at the time. A lot of these people, they were early on in their careers, and the only way we could, you can relax one constraint, you can relax the fact that they are talented, or relax the fact that it's apparent that they are talented. And we, not consciously, we relaxed the latter.

Patrick Collison: Finding the right people, you have to think like a value investor right, you're looking for the human capital that's significantly valued by the market. You probably shouldn't look to hire your friends from Facebook, and Google, or whatever, they are already discovered, and if they want to join you, that's great. They are probably harder to convince. John has spent a little while yesterday afternoon trying to figure out in retrospect, what traits out first ten people had in common and felt were significant. Generally speaking in culture I want to carry out everything we say, advice is very little advice, wildly extrapolated and I think there is a lot of truth to that.

For our first ten people, the things that seemed to be important, they were also very genuine and straight. And I think that matters quite a lot, there are people that others want to work with. That there are others that people trust, that they are intellectually honest on how they approach problems. They are generally people who like to get things finished. There are a lot of people who are really excited about tons of things. Only some of those are excited about completing things. There is a lot of talk out there, like hiring people off their GitHub resumes, that doesn't really ring correct to me as there is a large premium on lots of different things. I think it's much more a priori, much more interesting to work with someone who took two years to spend time going deeper into an area. And then the third trait that we looked for is that they cared a great deal, it's offensive to them when something is just a little off.

In hindsight there were all these crazy things that we use to do that, do in fact, seem crazy. Like I should have not done them. Everyone was always like, it was borderline insane how much they cared about tiny details like we used to. Every single API request that ever generated an error went to all of our inboxes and phoned all of us. Because it seemed terrible to get an error that didn't get a resolution from the users standpoint. Or we used to copy everyone else on outgoing email and point out slight grammar or spelling mistakes to each other. Because it would be horrible to ever send an email with a spelling mistake. Anyways, those are the three traits we came up with, genuine, caring a great deal, and completing things.

Ben Silbermann: I have something to say, I don't think there is a wrong place to find people. So when I look back at our first hires we hired, they came from all over the place. I put up ads on craigslist, I went to random Techtalks, we used to throw weekly BBQ’s at the office, bring your own food and drinks and then we would just talk to folks. I think every time I went to get coffee at Philz, one of you guys were recruiting. Because your office was strategically placed next to the best coffee shop. But I think the really good people, generally are doing something else so you have to go seek them out instead of expecting that they are going to seek you out. Triple when no one has ever heard of or is using the product that you work on.

John Collison: Yeah and it's probably really important to have a great elevator pitch, not just for investors but because everyone that you run into right now is six months to a year down the road a recruit. So the right time to have gotten them excited about your product, the right time for them to have started following us, is as soon as it can start. It's going to take a very long time to recruit people, so getting people consistently excited about what you are doing will pay back later.

Patrick Collison: Maybe this is a little tangential, but a bunch of our friends started companies right out of school. And we started thinking, what goes wrong in those companies? And I think the most common failure mode was doing something overly niche, overly specific or bad. I think there is a major shift in time horizon as you go from classes to a startup. A class plays out on a quarter, or a semester, where a startup is a five or ten year thing. And I think this is really problematic, because it's really quite hard to hire people for niche things. If you told someone, look we are going to build a rocket that goes Mars, that sounds almost impossible but sounds fucking awesome. It's really easy to convince people to work on it. Instead of, well we are going to work on, I'm not going to give any specific idea, probably going to sound like we are doing a startup. But if you pick something pretty narrow, generally comes out of this class project that's actually much harder to hire for.

Sam Altman: One specific specific question that has come up a lot is how, as a relatively inexperienced founder, you identify who the really good people are. So you meet people at these BBQ’s, you are friends, maybe you have worked with them. What did you guys do specifically to identify that this person was going to be really great? Or did you really get it wrong? When did you learn you could identify raw talent? Or say they work at Google, or Facebook so they must be good.

Ben Silbermann: You will never 100% know until you work with folks. So the flip side is, if the person you hired is not a good fit, you owe it to them and to the company to tell them where to improve and if they aren't working out, then to fire them. But I think, generally, the question of talent falls into two big buckets. One is, you have some sense of what makes them good at their job. And there are some areas where you can test that area. And there are some that you don't. And the ones where you don't are much more difficult. So what we would do is a few things.

Before we talk to anyone, we try to figure out what exactly is world class in that discipline need. And this comes in a little later when you are hiring the head of finance and you know nothing about finance, except what is contained in a library book that you got. Like an introduction to finance or marketing. So I always made it a habit of mine to talk to people I knew de facto were world class and just asking them, what are the traits you look for? What are the questions you ask? And how to find them? If you are looking for the next person that is as good as you, where is that person working right now and what's her phone number? I think that learning what's good and bad during the interview process is extremely expensive. It is an expensive use of your time, expensive use of everyone else's time. A recalibration of that really matters.

Then once you have someone in an interview process, you will build the process over time to screen quality. Pinterest, we have an evolving set of questions that are rotating through and we are always asking if these are good indicators or bad indicators of quality. The other thing the questions are supposed to do is give us a sense of, is this the right place for this person to come in and work? This is the point you guys made about being very transparent. About what's going to be easy or hard. Really great people want to do things that are going to be hard. They want to solve tough problems, so there was a sense of brilliance in Google sending out these interview questions that were thought to be really difficult. Then people who like solving problems, they come out and seek those. I think it's really important as companies get bigger that they don't whitewash the risks. I heard that Paypal, you go in and after the interview they say, by the way, Mastercard wants to kill us and you will be doing something that is illegal, but if you succeed you will redefine payments. Or when they were recruiting for iPhone, they didn't even tell people what they were doing. You won't see your families for three years, but when you are done, your kids, your kids' kids will remember what you built. I think that's a really good thing in recruiting as well. Be very very transparent on why you think it's a great idea, but you lay out in gory detail why it's going to be hard. And then the right people select in or they select out of that opportunity.

Patrick Collison: Evidence suggests they were able to see their kids though.

John Collison: I think one thing you have to do when identifying talent is have the confidence to interview in a way that works for you. I think, say you are not the world's best engineer and you are trying to interview engineering candidates. I thinks it's tempting to co-opt what everyone else does, get them to put things on a whiteboard and do other engineering things. In the case of Stripe, we flew a guy out and we spent a weekend coding with him and looking over his shoulder. It was the only way we could tell and get ourselves confident if that guy was any good. And I think you can extend that to any roles you are not an expert in. In that I am no business development guru but when we hire for business development roles, we have a project that we have them talk about, how they would improve an existing project that Stripe has or which new projects they would go out and do. And even if it is not my domain area, I am confident enough I can judge those really well. I think people often have this imposter syndrome when it comes to interviewing for roles.

Patrick Collison: I think a specific tactical thing to do, again, for the first ten people is to work with them as much as you can before committing to hire them. Once you hit a certain scale it's kind of impractical to put them on that side and be unskilled. Expensive from your side. But it's really worth it to the first ten people, right. In the majority, the first ten people, we worked with them to some capacity for a week in advance. It's pretty hard to fake it for a week, it tends to be quite clear quickly. Another answer I thought of to the question, how do you know if someone if great? And people talk about this notion of the 10x person or what the skill set is. I don't know what 10x means. I think the slightly more intuitive decision is, is this person the best out of all of their friends at what they do? It's a little insensitive on how they choose among friends, but for me at least I find that a better way to think about it is, is this the best engineer this engineer knows? And the other thing worth mentioning is, on the first ten people on the culture and team topic, I think everyone doesn't realize until they go through it themselves, how important it is because in life and media people focus too much on founders. Here we are and we are reinforcing the structural narrative that Stripe is about John and Patrick and Pinterest is about Ben. When the vast majority of what our companies do, 99% are done by people that are not us, right? It's obvious when you say it but it's very much not the macro narrative. These are abstracts and you associate them with certain people. For companies like Apple and others, Steve Jobs was a tiny, tiny part at the end.

John Collison: So don't screw up is what you are saying?

Patrick Collison: Something like that.

Ben Silbermann: I think referencing people is really important Referencing people is just what it sounds like. Asking people with experience for their honest opinion. We do that really aggressively but we are trying to figure out what this person is like to work with. We are trying not to validate if they told the truth on their resumes because we assume they told the truth. So a very standard question is in an interview I might say, hey we both know Jonathan, because we are both friends, if I asked him what you are best at, what you are the most proud of, or what you were working to improve, what would he or she say? Because it creates social awareness and accountability. Then I typically ask something that makes the question, that is typically soft, feel a little bit more quantitative and then calibrate that over time. To evaluate this person's dimensions, is this person the top 1% of the people you worked with, the top 5%, and the top 10%? And it forces scarcity that gives them material reference. Instead of saying, hey what's the best thing about John? You say, he told me he was good at these things. Can you validate it? Yeah, sure. It's the type of tool that should be taken seriously.

John Collison: And referencing, obviously, isn't easy to begin with. But it does provide really useful over time. And I think for name references people do want to be nice so you have to create an artificial scarcity by saying, where would you rank this person with people you worked with. You should aim to spend fifteen minutes on the phone with that person instead of letting them say, yeah this person is awesome.

Sam Altman: Also those references are a tremendous source of recruits. Once you hire these first people and they join, what have you done to make them effective quickly and to get them to the right culture place? Hiring is usually difficult but then not as difficult as making them happy and effective. So what do you do with these early employees to accomplish that?

Ben Silbermann: Well the answer to that has changed since we went from small to bigger. When we first started it was because we needed that person, a long time ago. So their whole onboarding was, here's your computer, we already set up your environment, don't worry about it, this is the problem we have to solve together. That's the nature of the startups, we were all in this tiny two bedroom apartment. All the other things, building personal relationships, spending time together, it all happened magically. You didn't have to do anything.

The one thing I would like to add to that is we always reminded people where we want to go with that someday. Because it's really easy to drop someone into a problem and they would think the whole world was this little problem in front of them. We would always say, someday we want to do for Google what they did for search. Our plan for trying to get that done.

Now as the company grows, I think that problem has to get a little more formalized. So we spend a lot of time thinking and constantly trying to refine what that person looked like from the day they came in, to their first interview, through 30 days after they joined. Do they have someone's name they know? Do they know who their manager is? Have they sat down people on their team? Do they know what the general arch of the company is? And what the top priorities are? And we have a program we do. It's a week long and they have functions to go in deeper. And that's something that has always been refined. And the output metrics on that is one, we ask people, what did you think afterwards, then 30 days afterwards? Then we also ask their peers and manager, hey is this person up to speed? Do you feel we did a good job at making them productive? If we haven't then thats a key that a) we should not be hiring any more people because we're not doing a good job bringing in new people and b) we need to retool that.

I think those things are important. I just wouldn't discount how important it is to get to know the person as a person. What's their aspirations? What's their working style? How do they like to be recognized? Do they really prefer being in total silence? Are they a morning person or night person? Knowing those things, it just demonstrates that you care about them individually and collectively, what your goals are.

John Collison: I think there are two things that are important at any stage, though the implantation will change. First is to get them up and running quickly to do the work. That is how you are going to find the problems, it is how progress is measured in the real work they are doing. And so when we have engineers start, we try to get them committing on the first day. When we have people in business roles start, we will have them in real meetings the first day on what they are meant to be working on. Sometimes it's easy to be tentative and ease people in. We are much more, push people off the cliff. Then second, we try to quickly give people feedback. Expectedly giving people feedback on how to adapt to the culture. When you think about it, if you have built a strong culture as all the companies up here are trying to, it's going to take some adapting from the person, it's not going to be necessarily easy. One thing we have at Stripe is the culture is a lot more written. So you have people next to each other, with headphones on, IMing each other. And for a lot of people coming in and working in an environment like that it's sort of hard-

Patrick Collison: In from normal places.

John Collison: Exactly yeah. So from everything high level of how you are doing at your job to minor cultural issues, the more feedback you give them, the better they will do. Its unnatural to be telling people if they are doing a good or bad job. You don't do that in your normal life, hopefully you are restrained. But when you have employees that is what you owe them to do well.

Sam Altman: So I think this is a good transition to when you're companies have scaled. What are the biggest changes you have had to make to your hiring policies and to how you manage the teams as you have gone from two to ten to a thousand employees?

Ben Silbermann: There are a lot of changes. I think one thing we try to do on the team side is to make the teams feel as autonomous and nimble as possible within the constraints of the organization. That means over time we are trying to make it feel like a startup of many startups. Rather than this model setup with form policies cut horizontally through it. It's easier said than done, I don't think we are all the way there but one goal is that each team has control, to hold the resources that they need to get the goal done. They know what the most important thing is and how it's measured. That way the management problem because somewhat tractable. Otherwise it feels completely impossible if you can't decompose it into atomic components. You just look at it like, Oh my gosh, complexity level is rising geometrically, on a management level, it's never going to work.

You have to create these abstracted units. At least that's what we are going to try to do. Pinterest in particular, the real challenge with building those abstract units is, we want units that encompass a super strong designer, or a super strong lead engineering, a writer, often times a community leader. We want them to be self contained. That kind of makes it hard, but that kind of core is to our philosophy to build products. We put people together that have all these kind of disciplines, lots of things then we anchor them to a certain project then we try to remove barriers to let them go fast. Then we find no barriers and we sit down to figure out how we speed that up. I think hiring is a little different. I think the biggest change and the biggest asset you get is the people, referrals become more and more the life blood depending on the network of the people you bring in. So one of the lucky and in hindsight decisions we made was our actual fourteenth or fifteenth person we hired was a professional recruiter. She worked at startups, she worked at big companies like Apple. But she sort of knew where that pipeline breaks down. Knew the early indicators, and taught everyone not just how to screen for talent but to identify the people who are going to be culturally really good for the company. And I think looking back on that, its something I personally really value.

Patrick Collison: There is a huge amount of stuff here that is under management growth. Either your company fails really quickly or all of your problems become about management growth. One thing that tends to take people by surprise, and took me by surprise, is how quickly the time horizons change. In your first month, you are largely thinking about things one month ahead, right? Maybe that is what your development road map is oriented around. Who you are working with, maybe it's a very informal relationships where they haven't fully committed to be full time or not. The more time that goes by, I feel that has a reciprocal in the time horizons.

In one year, you are thinking a year ahead. After four years you are thinking four years ahead. That increases very quickly right? After one month it's super short term. After eleven months, you should actually be thinking and planning a year ahead and think about human structures. Thinking about stuff Ben talked about. Where you want to be going long term. Things like that. I think that plays into the hiring and that in the early days you have to hire people who are going to be productive. Essentially , you don't have the luxury of hiring people that look to be promising but they are not going to be up to speed for another year or two. They have to be able to work immediately. But after two or three years, then it becomes much more reasonable to make those investments.

Actually if you are not making those investments you are probably being much too short term. And so I think that's really important. All of these problems in some sense are easy. Like, how do you build such good bonds with people? We all do it every day. How do you make it systematic, and effective at scale? It's all going to vary significantly, in perfect proximation of what you would ideally do small, what hacks can you pull? Make it work as well as you possibly can at a large scale. A rapidly growing company, with head count here two-three heads a year, it's a very unnatural thing. What's the least bad way of managing that growth? I think it's worth being systematic about it, thinking of ways to do it. Realizing you can't do much better than ask a question. For Stripe, it's things like, we have three meals a day at long tables where everyone can sit together, if you think about how much more human interaction happens with having these randomly mixed meals. It's vast, right? A whole list of things like that. I think that's the general framework.

Ben Silbermann: One thing I'm really curious about. You guys value transparency. Have you scaled it over time? I know with us, we think about it all the time. Just curious.

Patrick Collison: Startups, I can't remember who defined it as, a startup is an organization that is not yet stuck with all these principal agent problems. That large companies, what is locally optimal for you frequently is not what is globally optimal for the company. As a consequence of that, a reason a startup can just work differently than a big company, at a big company, a lot of the things that are good for you, you couldn't do them in a completely transparent environment because people would think less of you. But because everyone is rolling in the same direction, a startup, you can kind of make all the information transparent. Like I said earlier, Stripe used to bcc us to be on every email unless you opted out of it. We thought that would be more efficient, you wouldn't need to have as many meetings if you could keep abreast of what is happening. And over time, we sort of built an interesting framework of mailing lists. We now have a program for generating gmail filters. For a rocky path of fifty people or so to Ben's point of asking people how they are getting along after several days. They all reported terrible because they couldn't find all the emails people were sending to them. They were missing things and everything.

John Collison: Gmail broke at one stage.

Patrick Collison: Right, right. At one point Gmail broke because we were sending too much email. It is hard to scale because you might contact somebody out of the company with some great idea. The person sitting across the way from you thinks it's the stupidest thing they have ever heard. You are kind of under that scrutiny of the whole organization in some degree with all your communication. That's the challenging side, that people more formally know what's happening. I don't feel I can give a stronger endorsement of it than it has worked so far. I really am curious how it will work when we are five thousand people, if we are ever at that scale.

John Collison: I think a couple things that helped us scale it, is we changed tools in changing the culture around it. On the tools front it use to be the case you could keep abreast of what's happening in the company by reading all the email. Now we weekly all hands on deck. We actually have to put all this work into communicating what is going on in this company. Since there is so much more. The second is, cultural side so much information in terms. You have to create social norms around it. Obvious things such as what is confidential to Stripe. Less obvious things, like when emailing someone or talking in Slack or IRC that is now viewed by one hundred seventy people, it's pretty easy to get stage fright. And it's pretty easy with what you thought was a reasonable proposal, you get this drive by criticism and you are less likely to share in the future. We have had to create norms around when it is okay to jump into discussions and how that interaction works because people are around the stage much more.

Patrick Collison: I'm sure it's not good to put anyone on the spot, but Emily interned at Stripe this summer. I am curious, as an intern. what you thought of it.

Emily: Overall it's great. The first week, I spent most of my time reading Hackpad trying to get caught up on what the company was doing. It can often be quite distracting from your own work as often times there are other parts of the company that you are interested in.

Patrick Collison: Hackpad, by the way, is like Google Docs but with a news feed, where you can see all the documents.

Emily: And you are encouraged to make everything public, everything that you work on. But overall it gets you sped up rather quickly. There are these things called Spin Ups, where every leader of the team gives a thirty minute talk on what they are doing and how you can contribute if you are interested.

Patrick Collison: Do you think transparency was that good?

Emily: Yes, I remember having a hard time remembering what I should and should not subscribe to. The first week having two thousand emails in my inbox, then by the end there are three or four teams you want that information coming up from.

Sam Altman: So this question is for Patrick and John. Have the people that you hired early been able to grow up into leadership roles?

John Collison: In Stripe's case yes. A lot of the first ten people are in leadership roles now. I think that's one thing that corporations, it's an unnatural skill that they need to get good at. People don't exactly come out of the womb being good at management or at leadership. And being able to develop that in people and helping people progress as they spend a number of years at the company, it's a lot of work when people are running around with their hair on fire. But it's also damaging if the company can't develop that skill.

Ben Silbermann: I think for us the answer is some yes and some no. I think one of the benefits of working at a startup is you can be handed a challenge no one else would be crazy enough to let you take on. And that could be managing people, it could be taking on a project. And also if you ask someone to take a risk like that, it shouldn't be one way through the door if you don't succeed. Otherwise it creates fear to give it a shot. So we have some folks managing a large team at the start, individual programmers, individual engineers. And they say, Hey I would love to try leading a project, leading a group, then taking responsibility for management. Then we have other folks that try it and are really glad they did so they know they never want to do that again. We try that, for those people, you can have just as much impact on the company through your individual contributions as an engineer or what have you. But it's really hard to predict unless you give people a shot. So my strong preference is you give as many people a shot as possible. And in the few areas where you feel there is too much of a learning curve relative to the business development you are trying to achieve, that is when you look for someone who might walk in and really execute well on the job.

So the question was, has the vision changed since we initially started? Well I think on the vision, when we first started hiring, we were like we are going to build this really cool tool, people are going to enjoy it. I like collecting things, maybe others like collecting things. And what we didn't expect that revealed itself early on, was that looking at other peoples collections was, it turned out to be this really amazing way of discovering things you didn't know you were looking for. It becomes a solution to the problem that a lot of other technologies don't have.

Over the last year we have poured a lot of resources into building our recommendation products, search products, feed products. Leveraging the unique data that we have, which are these pins that were all picked by someone and hand categorized. And the on the audience side, I think the first big surprise was truthfully when we first started, we didn't know if anyone would really use it. eE were just happy that anybody who wasn't related to us or obligated would use it. The biggest surprise has been the diversity of people and how diverse those groups have been. And I think that's been one of the things that's the most exciting. And the funny thing is, often as the company goes further along your aspirations get bigger. There is this gap that exists, and I tell my team, between where we are and we should be. Objectively we are further along, I feel the gap has widened. But I think that's a really common trait among people who found companies.

Sam Altman: So the question is, most start ups are not the iPhone. You can't guarantee that most people's grandchildren are going to remember this because most startups fail. How do you convince people to make sacrifices to do join a startup?

Patrick Collison: I think part of why it resonates with people is because it's not guaranteed. If it was it would be boring. There is the prospect of affecting this outcome, but nothing more than that potential. As far as not seeing their families and kids, startups do involved longer hours in the beginning but I think that story is overstated. Even the startups that in the earlier days had some sort of longer working days, have a tendency to exaggerate. It's kind of like the startup version of fishing. Every startup thinks they worked more insane hours than the next one back in the early days. It's like we literally never slept for two years. I think realistically for most people, it's not that big of a sacrifice. I think on average, people work on average two hours more a day. It is a sacrifice but it is not forgoing all pleasure and enjoyment for the next half decade.

Ben Silbermann: Even the iPhone wasn't the iPhone before it got done. No smart person you are hiring thinks you have a crystal ball into the future that only you have and that joining is a guaranteed thing. And in fact if you are telling them that and they select in, maybe you shouldn't be hiring them because they didn't pass a basic intelligence test of certainty and the future. But I think it's fair to say what's exciting and where you think you can go. And where it's going to be hard and chart your best plan. And then tell them why their role in it can be instrumental because it is.

I really liked what you said, if you tell people, Hey we are going to go to Mars, it attracts the best people and you are closer to Mars and they know that. What I would discourage is whitewashing all that. And if people are joining because they want all the certainty of Google and the perk of working in a small startup with more email transparency, then that's a really negative sign. For example, when I interview people, they often say, I'm really passionate about what you are doing. I often ask where else they are interviewing. If they list seven companies that have nothing to do with each other, except they are at the same stage, I love the stage of discovery, so I'm interviewing at Stripe, Jawbone, Airbnb, Uber, I'm also putting my resume into Google X, that's a sign they are probably not being authentic, which you care about. And those folks, when things get hard, they won't stick it out and work through it, because they were really signing up for an experience, not for achieving a goal.

Patrick Collison: I think the other thing that motivates people a great deal is the prospect of affecting some outcome, is just the personal development angle. And a startup because it's more lightly staffed it's much less forgiving. Whether or not you are the best or the worst person in the world, you are probably not going to alter Google's trajectory. Whereas if you really want to benchmark yourself and see how much of a contribution and impact you can make, the startup is a much better place to test that.

Sam Altman: How does your user base affect your hiring strategy?

Ben Silbermann: Conventional base, you only hire people who use your product religiously everyday. And that probably works well if you make an API. For us, we screen for people who have vision and discovery online. And they have to know how our service works, and they have to have used it. But they may not be a lifelong user. And for us that's great, we can ask what is the barrier that is preventing you from using it? Come join, we will move that barrier. Help us get closer to that vision. If you read a startup book, there is all this wisdom, but it is only useful if it works in your certain circumstance. So for us, we have had to broaden the lens a little bit and bring in people who are excited about the mission, that care about our product and our approach to building products. Even if from day one they weren't our earliest users.

John Collison: The one thing I want to tack onto this is, we touched on it being hard to hire early on for those first employees, you have people with other options, you are very much at the ugly duckling stage. Hiring people who are passionate about your product is a great way to find people. You have a natural advantage over other companies. I know in Stripe's case, we hired four Stripe users, people who we probably couldn't have gotten otherwise. I'm sure it was the same in Pinterest's case where you will get all this benefit at working with Pinterest, like, Hey it's Pinterest.

Sam Altman: Thank you guys very much for coming in today.

Lecture 12: Sales and Marketing

Aaron

Can we keep playing, yeah.

[Eye of the Tiger starts]

Can we turn it up a little bit so it has more pump up? Okay, there we go. Okay. Guys, we have to find the beat then clap to the beat. Okay, please stop the music. Please put on the presentation. Thank you. That will be about the most pumped up thing that happens in enterprise software. The rest is downhill from here. Thank you for that well rehearsed intro.

I’m Aaron Levie, CEO and co-founder of Box. Welcome to this edition on how to build an enterprise software company. This is my understanding, this is the course you are taking? Is that correct? No.

So this is my job today. I am going to try and convince you that everyone else that speaks during this whole class is wrong and that you actually want to build an enterprise software company. Hopefully we will be able to work through this and you'll have a good sense of why it's super cool to be in enterprise. And why the perceptions of going into the consumer space, why it's so much fun are wrong, and why you want to go into enterprise software. Who wants to build an enterprise software company? Good, alright. Thank you very much. Hopefully we will do a vote at the end and hopefully that will not have shrunk. That's really the only goal I have today.

So we are going to talk about three things today. The first is the quick background of Box. Because when we first started out, we did not know we wanted to do enterprise software. So I want to go a little bit into why we went after enterprise and what we do today. Then we are going to talk the major factors that changed in enterprise software that make it possible to do a startup today. And finally we are going to look at patterns that are ways to recognize and go build a startup by yourself. Hopefully, that will be some practical, useful advice. Just as a forewarning, my voice, I've been speaking a lot the past few days. So hopefully I will be able to get to that third part of advice and make it.

Building for the enterprise, these are high level stats of Box. We have about two hundred forty thousand businesses that use Box, there are over twenty-seven million users that have brought Box into their organization, and ninety-nine percent of Fortune 500s. Actually that one percent is really Microsoft and they don't seem to want to buy from us. We have to work on that a little bit. A lot of users to bring Box into workplace environments, these are some of the organizations that are using the product. We have a very wide range of industries from manufacturing consumer products to companies like General Electric. Stanford Health Care actually uses the product for collaboration inside the medical department search. Between health care, media, manufacturing, these are some of the range of industries we serve.

So the question is, how did we get here? Because we didn't start the company to be an enterprise software company even though that is how things ended up happening. We launched the company in 2005, we got the idea back in college which was 2004. Was anybody using the internet back in 2004? Okay great. I didn't know if millennials used the internet or not, so great. Sorry, okay no more age jokes. Okay here is the point. So back in 2004, you might remember there wasn't a lot to do back then. It was boring right? This was before Facebook, this is certainly before Snapchat, so before much to do. You couldn't send people fifteen second messages or photos that disappeared because you didn't even have phones. So on the internet, in 2004, there wasn't a lot going on. This is sort of what the internet looked like, a barren deserted landscape. Just to clarify, the happy camel is Google, the sad camel is Yahoo! This is the internet in 2004. Yahoo! has done a lot better since then, but back in the mid 2004, they were trying to find their way. And Google was taking over the world. But this was the extent of the entire world.

So what we noticed in 2004, in college, was for some reason it was really hard to share files. And as simple of an idea as that is now, and you go back ten years. It was either really expensive or really hard to move data around through corporate companies. I had an internship at the time, most of my job using data was to copy printed out papers and put them in cabinets. That's what you do as an intern if you are not a computer science guy. So I was really really good at copying paper, unfortunately not a skill really used today. But it was really hard to share files. In classroom environments, you were working in large groups, it was also hard to share files. I went to USC, and USC gave you fifty megabytes of storage space. Fifty megabytes, you can basically store one file. Then it would auto delete every six months. So whoever was running IT at the time, they certainly weren't running hard drives. And so it was really, really hard to store and share files. Well, why don't we make it easy to store and share files from anywhere?

So we got the idea for, at the time, Box.net. And what we noticed was, there were a lot of factors that changed in the software world. The first was the cost of storage was dropping dramatically. So in our business, basically every year or two you could double the amount of storage and data goes into a hard drive. So what was uneconomical now because feasible. The cost of computing, the cost of storage has dropped. We had more powerful browsers, and networks. Firefox was just emerging. People were using the much faster internet for homes and the classroom. Then people had more locations that they wanted to store and share information with. So we had these three factors that were sort of emerging. So pull these factors back when I give some tactical advice. The first point to remember, always look for the changing technology factors. Every market that has a significant change in underlying were enabling factors was in an environment that was about to change in a very significant way. We were very fortunate in the need for data in the Cloud, was growing in importance. The cost and feasibility. was also not improving rapidly. We decided to put together this really quick version of Box and launched it as Box.net. The idea was, let's make it really easy to share files. It turned out the idea clicked. We got angel funding from this guy named Mark Cuban. This was before Shark Tank but it was very similar. So we got this funding and thought this was going to be super exciting.

We are going to drop out of college, we are going to move to the Bay area, and it's going to be awesome. And when you drop out of college, anybody drop out of college yet? Okay, good. Stay in school! When you drop out of college, everyone pictures it like it's going to be incredible. Bill Gates dropped out of college, it will be like Bill Gates. Or Michael Dell dropped out of college, it will be super exciting like Michael Dell. Steve Jobs dropped out of college, so this is what people imagine, but nobody ever remembers that this guy dropped out of college also.

So it's not really a guarantee that it's going to be successful. It's funny, I don't even know if this guy dropped out of college. It just seems like he had to. And I apologize if anyone is related to him, it's just a funny picture on the internet. So basically we decided we would drop out of college, we moved up to first Berkeley, then Palo Alto. We decided we were going to open up the product for free. We got hundreds of thousands of people to sign up for the product every single month. If you go to Box.net you get one free gigabyte of file storage space. Which once again was big back in 2006. But we were getting so many users, we were trying to figure out what to do.

What we ran into was a common problem that, really, any startup runs into. Really pronounced by our business model which was, for consumers we built a very robust, very reliable enterprise. We really brought a really insignificant product. So for consumers, what we were running into was we had all these features you could pay for but a lot of consumers didn't need all those features. And for enterprises, we really didn't have enough securities and we didn't have enough capabilities around how enterprises want to use their data. So we had more than what a consumer needed and not enough that an enterprise needed. So we found ourselves at the juncture. We found ourselves basically in this period where its very difficult to figure out what we wanted to do with the business. So we had to make this choice. We were at this path, where we had to choose which path to go down.

This is back in early, mid 2006 up to late 2006. I was 23 at the time, my cofounder was 22. Our founding team was even younger, we had all dropped out of college. So in 2006, 2007, we imagined these two paths and the worlds were very very different. When you do a consumer startup it's basically lots of fun. You have parties all the time, it's just super exciting. Then in the enterprise you are battling these large, it's a rather thankless model because people just generally hate enterprise software. So that was sort of how we imagined the two paths, was we had to choose one of these two worlds. So we looked at that and thought. Okay consumer looks really fun, enterprise looks really hard and there is a lot of competition. At the same time, in this consumer space you are always fighting this issue of how do you monetize? How do you actually get people to pay for product? In the consumer space there are really only two business models that you can do.

You can either have people pay for your application or you could provide advertising on the application. To give you a little bit of perspective, these are today's numbers. In the consumer world there are about thirty-five billion spent on mobile apps every year. Pretty big number right? Thirty-five billion dollars. That's a lot of money being spent on mobile apps today. For advertising, the global digital advertising is $135 billion dollars. So most consumer businesses are going after, if you are not doing e-commerce, are going after about $170 billion dollars of either purchasing power on applications or global advertising around these types of services. So big number, a lot of opportunity there.

However, in the enterprise there are $3.7 trillion on enterprise IT every single year. These are the servers, the infrastructure, the software, the networking, the services. All of that stack of technology equates to a few trillion spent every year. What we realized was there was a rather wide delta between these two markets. We are going to be fighting to get consumers to pay a few dollars a month. And Google, Microsoft, and Apple will try to make this product free over time. And there were rumors that google drive was coming out. And all these products that were going to happen, are coming out. But in enterprise, its not about them trying to save money on IT.They are either trying to increase productivity, they are trying to increase business. So the value equation is very different. So the consumer, we have limited amount of money that we wish to conserve for as few things as possible that we are going to spend. On the enterprise it's a little bit of a shift, actually what can I get out of technology? How much value is that for me? So that was a really important data point.

However, the problem was that enterprise software was very unsexy right? Very competitive, very hard to build a business. It wasn't something you shot out of bed in the morning saying, I'm super excited to build an enterprise software company. And the reason for that was actually very straight forward at the time. The way that you built software was very slow. It was very slow because you couldn't break anything for customers, the sales process was very slow because customers take a long time to purchase technology. So I think everyone is used to this philosophy that when you are trying to sell enterprise software, it could take up to years for them to actually just buy the software. Then it could take even more years for them to implement the technology in the first place. So a lot of companies are around for years without their technology even used in the first place.

That felt like a huge problem, and not something that we wanted to be a part of. The technology itself is complex, I don't know how many people have had to use enterprise software but it's generally really complicated. You try to figure out, why in God’s creation did a designer try to put forty-seven buttons on one page. You just can't even understand it and the reason is something we will get into in a second. But basically there is just no love or care for the design or user service. The software is just complex. And finally, if that wasn't bad enough, you have to figure out how you are going to sell this software. For anyone who loves the power of the internet, this notion of having a sales intermediary to get to your customer, seemed really unappealing. You have to hire a bunch of people, who are going to be in every country, they are going to be the only interface you have to your customer. You hire this guy named Chuck, and Chuck is going to roll in with a brief case and he is going to try to sell lots of enterprise software to the customer. Just so we are clear, this is what Chuck looks like. And that was the sale process that you, in the enterprise at least, that we imagined in our head. Chuck looks like a happy guy, but he is still an intermediary to getting your software. Well why cant use the power of to internet and get our technology out here that way?

Why should we have to go through this sales intermediary as we scale up the business? I will get into it in a minute why we were wrong about the sales business. But this was sort of the fear that we had. And if that wasn't hard enough, we had investors saying, in 2007, basically there is no way you are going to make it in enterprise. You again, are basically a founding team of 20 year olds. You don't have anyone on your team that has been in an enterprise. Microsoft, Oracle, IBM, these companies are going to stomp on you. This is going to be very very hard to succeed. And to be fair they were right on several areas. We were a very inexperienced team. We were still very early in our careers. Our co-founder, for instance, looked like he was 13 years old. Just to be clear on what he looked like. So it sort of made sense right? This is him as our CFO, I think this is him at 29. But it looked like we were going to run off with the money and go to Disneyland. I appreciate why they didn't think we could pull it off. I can't imagine giving him money.

So, we decided that we still have to go do it. We have to give this our best shot. We are going to take the scale, the consumer experience, the DNA of our company and we are going to see if we can bring this into the enterprise. We were really fortunate. We had an investor, early in his career, make a belief on us because there was something changing with the enterprise that we would be able to take advantage of. We decided, if we are going to do the enterprise, if we were going to go after the enterprise, we were going to have to play with a very different set of rules. So what about the complexity of software can change in this era? What about the sales process is very slow can change in this new era? How do we move and go directly to the user the customer, instead of having this really indirect process at getting our technology out there? How do we build a design for the user instead of just for the RFP process that a customer is going to go through? So we looked at all of the factors that are true with the enterprise and we are going to do, not in all cases, the opposite. We are going to find what has changed in the technology world that we can build a newer, and better software company. That was the decision we embarked on, the path we embarked on 8 years ago. And that is why we have been focused on enterprise.

Today, again we have about two hundred and forty thousand businesses using the product. And the reason is we architected the business model, we architected the software, we architected the solution to work in one specific version of the world, and it turned out that one solution was the one that happened. And I will go a little bit into what has changed in the world that we sort of built our company around. And what I would highly recommend to you, if you are building an enterprise software company to orient to your technology. So that was sort of why we made the decision, how we started to take on the problem.

So everything about enterprise, and by definition the software that the enterprise uses, has changed just in the past 5 years. If there ever was a magical time to build an enterprise software company, now is that time in terms of how much has changed in what is going on with organizations. Lets go over a couple of these things. The first is that most application companies are moving to the cloud. And the biggest thing is, if you are going to start a business management company, or a business intelligence company, even a contact management company years ago you had to have you idea implemented in every single customer location. No matter how many customers you sold to, no matter what region you were in, every customer had to put that in their datacenter. That was the flaw with on premise computing. You were doing all this work, you were creating so much redundancy, it was the slowing down the whole process of delivering and building software for the enterprise. All of a sudden the cloud came around, things like Sales Source. com, things like Amazon Web Services, basically said. Why is it that every customer that wants to implement a couple servers, have to implement servers, put them in their data server, but security or networking around them, six months later they go live and a developer can use them in the organization, same thing with an application? They said, why does that make any sense today? We could just put together tens of thousands of servers, put them on demand, and you can use whatever you want, when you want and we can do that. That obviously is the definition of Cloud Computing. What's happening is CIO’s and large enterprises are taking advantage of this. So it seems obvious to everyone in this room because you would never build your company by buying your own servers. You would start is on google, yahoo, or ashore rather. But to an enterprise there are decades of infrastructure that now has to move to that cloud. So thats a massive shift that is actually happening.

We are moving to a world of cheaper, on demand computing from a world of expensive computing. The benefit of starting a startup is the customers don't have the same friction,, they are going to go and adopt new technology. As soon as the computing becomes cheaper, its easier to adopt new solutions. Which means, their barrier for showing you in--the barrier is a lot lower which is great for startups. We are going from a world of customized platforms to standardize softwares. It use to be that you had to build all the customizations, all the customer experiences on top of the software itself and now customers are realizing that they won't open platforms and they can customize a layer on top of the product. It use to be that when you started an enterprise software, you could only sell to the top five or ten thousand companies in the world. Because only those companies had the wear with all, the talent, the infostructure, and the budget to employ you technology into the enterprise .Today, literally a two person company can sign up for box, as well as we work with General Electric who has over 300 thousand employees. So the fact that you can now serve a small business anywhere in the world, as well as some of the largest on the planet means there are much larger markets you can go after. Which makes it an even better economical proposition to go after the enterprise. The platforms themselves are becoming more global. Our customers were internationally a couple weeks after starting the company. If you would have done enterprise the traditional way that would have take years to actually be able to go internally.

And finally, the most profound shift of all, mobile devices. iPhones, iPads, Androids, Tablets, IT of these models have become a lot more user led. It's fundamentally important. In an IT world, incumbents generally win because they have the existing relationship with the IT organization, with the CIO, with the spending power of that company. In a user lead model, users are bringing in their own technology. They’re bring it in in the sales team, they are bringing it in in the marketing team, they are brining it in in finance and you can build software around that user. Which means they can bring the enterprise in and you can sell to the enterprise when they want to have better control, better security, better scalability.

So you still have the same model as a business software company but the way to get into the company now is through the end user. So those are quantitative factory changes. Just a couple quantitative changes, there is over nearly 2 billion smart phones on the planet. That changes every single IT model planet. Because it use to be 10 years ago, if you were managing technology for the company, you just had to manage the computers network that was inside your building. But now with billions of smart phones you have to manage ways of computing anywhere at any time on any network. And that becomes big in software companies, because no incumbent has built a technology stack that powers this line of work and how enterprises are using their data. So that creates a massive start up opportunity.

There are nearly 3 billion people online. That means that every single enterprise is changing how they are going to give their own products to their customers. Which means that every industry changes. There are only two times, two moments of opportunity where a technology revolution will happen in an enterprise. The first is where raw materials change. So cost of computing goes down and they centralize and let people use it on demand. The second thing that can change, is the very people that these enterprises have to go after need new experiences at that enterprises product. Let me give you an example. If you go off campus you probably use something like Uber or Lyft. If you are in the shipping business, if you are in the transportation business, Uber represents a massive change to your industry. So you can't let Uber exist without understanding, what are the implications of Uber? What are the implications of Instacart? What are the implications of Lyft to my business model?

So in a world where enterprises are dealing with that kind of change, you are going to need new technology to help them create their business models, how they adapt to this disruption. This is why it's such an amazing time to even start vertical software companies for industries. Right now every single industry is going through a business model and technology orientated disruption. Means they are going to need technology from start ups to help them work through this. I will give you a couple of examples: so in the retail industry there's this vision of omni channel or multichannel commerce. You are going to shop online, you are going to shop on your phone, you are going to shop in a store, and you want things to be delivered to you as well. So most of the incumbent technology does not power multichannel commerce. No one is prepared, what does it mean when consumers want to go buy goods anytime from anywhere with better information, better intelligence. So every retailer in the world is going to need a new technology stack to power their retail experiences.

In the healthcare space, every single health care institution is trying to find ways of building more personalized experiences, more predictive experiences, they want to have medicine be adapted to the individual. As the business model of healthcare changes to being about the surgery, charging for the check up, and instead, really where the customer pays for the wellness and staying healthy. Then all of a sudden every healthcare institution needs technology to deliver health care experiences. They are going to want to deliver telemedicine. They are going to want to deliver health care in more regional locations instead of just in the monolithic hospital environment. There are going to be new use cases coming around. How are our healthcare providers get connected to one another so one doctor can make better decisions? All of these things are going to require new enterprise software to power these businesses and industries.

In the media space, as an example, you have a world where the industry is going from really linear programming, whether that's television or that's music or that's movies, it's a linear supply chain oriented business model, when a film gets made, it goes to the theater for 3 months, then afterwards it goes to iTunes or other platforms, to a world where people want experiences on demand. So that's going to change how distribution works on a scale of 3 billion people on the internet, and again no media network has a platform that is actually going to power how content, data, and information moves to this system at scale. I was just in LA yesterday, meeting with a media company that has basically done predictive analytics to find their potential moviegoers in the middle of 3 billion users. They want to be hyper targeted on how to get to the specific 3 billion fans that are into a certain film types. And so all of a sudden you have a movie company who needs big data and they need business intelligence and marketing in order to go and market and distribute.

This is where two industries come together, where all new software is going to be necessary. So every industry is going through some form of this change. You can take any industry you want and zoom into it and say, what are the underlying technology factors that are going to change the business model for the next couple of years? And then there is going to need to be software to power those types of experiences. Think of the future of water, who is going to power that? That's going to need software, I'm sure.

So basically every company in the world, the great thing about being at Stanford is we study the technology. And we think of the technology industry as an industry. But in reality what is happening, is every industry is going to have a technology component of what they do. Enterprises are not going to be able to survive in the future if they do not get good at technology. If they don't have competency at leveraging data and using these new tools. But they are going to do that by working with what we call the technology industry.

Instead of everyone else building out in these expertise for themselves. So there's going to be a lot of partnership over the next five to ten years where companies are going to need technology to work smarter, to work faster, they are going to need to do this more securely. And this is not only going to change how individuals work in these environments but ultimately change the business models of these companies. So that was chapter two of this.

Now I'll give you some practical advice to help you get started. To be fair, most of this advice is looking through the lens of retrospect which means this is not how this is going to happen, but I can look back through time and say that these are the things that led this to be true. It's hard to be deterministic about building a company. You may not have all these things figured out, but this will give you a sense of pattern to recognize as you are building or thinking about building an enterprise software company.

So the first one is spot technology disruptions. This is going to be true whether you are building consumer or enterprise. The rest are more enterprise, but this is just fundamental if you are going to build a tech company. You have to look for new enabling technologies, or major trends, like fundamental trends, that create a wide gap between how things are done and how they can be done. Looking back in time to our business, the gap was basically storage was getting cheaper, internet was getting faster, browsers where getting better yet we are still sharing files with this very complicated, very cumbersome means. Anytime, between the delta of what is possible, and how things work today is at its widest. That is an opportunity to build new technology to go solve a problem. As you are looking at the enterprise, the question is, what about the cost of computing dropping so rapidly changes what enterprises can do with their data?

What does it do to change from a business standpoint? What was impossible, because of either economic feasibility or technical feasibility that 10-15 years ago is now possible. A fun thing to do every now and then, if you find a newspaper article from the 1990’s or 1980’s, business articles about technology, all we are really doing is repeating all the technologies we tried 10, 20, 30 years ago. It was too expensive, too unusable, and we didn't have the enabling technologies to make it possible. You can see this concept emerge, something that was impossible 5, 10 years ago is now very practical. I will give you an example. There is a company called PlanGrid, does anyone know what PlanGrid does? Okay, cool. Are you in the construction industry? You are? Oh my god! What does that even mean?

Q: PlanGrid? Or construction?

Aaron Levie: Construction.

Q: I work at a job site, we build buildings.

Aaron Levie: Holy crap, that's great. Basically PlanGrid is a mobile application that lets you manage construction projects, lets you access your blueprints, lets you manage all the data around a construction process. And what this company realized is, 4 billion dollars, I think, are spent every year printing out blueprints. And they have all the prints and updates to them anytime there a change, then they have to ripple and cascade through a very wide network of contractors and construction workers every time. Even if it's one slight, minute change, suddenly they realized, with the iPad, we have the perfect form factor to load up blueprints and content. This is something that can ripple through the construction industry, which isn't really known for high technology, except on the design side. How can they build technology that makes data collaboration problem really seamless and easy to do in an industry that hasn't really changed in a while? It was a perfect discovery of a change in a market and figuring out how those two things converge. Then this team built a great startup for it that is doing incredibly well and taking over the construction industry as proven by this individual. Thank you.

The next thing is, in enterprise, you want to start intentionally small. What I mean by that is you want to find, this is more true with all companies in an enterprise in a user light paradigm, you want to find the wedge that is sort of natural that you can create a product that will slip in the gaps of other existing products. But something that you think over time expands to be a more important product of the enterprise structure. What you want to start to do is say, we will take this sliver of a problem, we are going to make the user experience on the incredible. We are either going to change the business model, we are going to create new technology to make this previously problem really really simple. It might feel small at first, maybe you are going after small businesses and then you are going to go up market. Maybe you are starting with a sliver of the use case and expand out, but you intentionally start small. Because you will not be able to compete with an incumbent because the incumbent is always going to go for the full solution. So you have to find, what are the gaps in the full solution, that are significant enough that the customer is going to want to solve the problem with a discreet technology. But over time you are going to be able to expand. Again to either larger customers or to more use cases over time.

Great example is ZenPayroll. ZenPayroll was started by Stanford graduates couple of years ago. Basically, they discovered that the payroll is some small business is complicated and incredibly annoying process. That is because we use the same vendors that we have for decades to do that and they were digitally ordinated. You didn't get your payments as a receipt over email. Very complicated. You didn't get to see graphs of your salaries. There was really no good data around this. And they said we are going to take off the slice that is most painful to start out at, around hiring people and paying people. Just that payroll management process. We are going to plug into a lot of existing structure. But we are going to make it dead simple to go do this. And now they are able to move up market over time as well as deliver new services over time. And what happens is the incumbent in this market, eventually looks at something like Zen Payroll and says, well thats small. Its only for small businesses’. How can it be very powerful? But thats just the start. As they get that wedge, as they fit into the market, they are going to be able to expand again over time. Build out more services and more capabilities. But they found just the right, exact opening to build a new company and have the emerge.

Then next you really want to find asymmetries. You want to do things that incumbents can't or won't do because either the economics don't make sense for them, the economics are so unusual, or because technically they can't. I will give you two examples. So, if you are going to build software today for the enterprise that goes after an incumbent category, that has more of a suite? oriented approach. Then what you are going to want to do is build technology that is platform agnostic. What suite players will do is want everything to be integrated with itself, and theres more value with the vertical integration. But you want to go after a different access. Which you want your technology to work across all the platforms. That way you can work with so many different kinds of customers. You can be an ally to so many kinds of platforms, which a traditional incumbent is not able to do. That is technically infeasible because its architecture or its a fundamental component business model to not do that. The other thing is, trying to do things that is economic feasible. You can look at the cost structure of an incumbent company and discover where they are not going to be able to drop their prices. Because that business model is fundamental to the company. Or where can you find ways of monetizing the customer that are unusual or unique that no one has discovered before, thus making impractical for anyone else to do.

There is a company called Zenefits where they have an HR management software company that helps you as a small startup manage all your benefits, all your HR information. And instead of charging the startup that may not value the software stage they are at, they realize they can get commission from the insurance companies that pays for the ability to use their software. The customer itself is not paying for Zenefits. Zenefits platform is being paid for by the insurance company and they have thus created a business model that no other software company has been able to think of or attack. And they are equally going and disrupting a category that has not seen a lot of innovation previously which is the health and benefits space in small businesses.

The next is you want to find the mostly crazy, but still reasonable outliers within the customer ecosystem. So you need to find the customers that are at the edge of the business, their business model, their industry and find the unique characteristics of those customers. Leverage them as your early adopters. Paul Graham has a great article where he talks about living in the future and building what is missing when you are living in the future. Thats an easy way to spot trends and patterns about disruption that is playing out. The same is true in the workplace. If you find customers that are working in the future, you will be able to work with them to find what is missing in the future. And how do we build technology that supports all these new use cases that are going to emerge? There is a company called Skycatch that does enterprise drones. At first it seems kind of weird, but in construction space, farming they are using drones now for data capture and modeling different environments. So this company is able to find all the companies that are on the bleeding edge of their industry. What is unique, or new about how those businesses operate. And they worked with a lot of those early adopters to establish their platform. Which really is first enterprise drone company. So the idea is, go look at your market. Find the customers on the bleeding edge of their market who use technology to get a head. And that use technology for performance advantages, and go work with them to see how your product can evolve.

Listen to your customers but don't always build exactly what they are telling you. This is a really key distinction around building enterprise software. Your customers are going to have a large number of requests. Your job is to instill those lists down into the ultimate product. This does not mean that you are not going to build exactly what they tell you to build. It is your job to listen to their problems, and translate those into what is going to build the best and simplest solution for them. It's really your job, and Palantir does a really good example of very very complex issues and then scaling them down into simple solutions for complex problems that the customer would not have known how to ask for.

You want to modularize not customize. So build a platform as opposed to building all the custom technology and customer vertical experiences into the software itself. Make sure you really think about openness and APIs as a way of building experiences. Don't build that directly into the product. Focus on the user always. The magical thing about building an enterprise software company right now is you can keep consumer information at the center of the product. That will always mean that adoption is easier, that your product has a much better chance of going viral. It becomes easier to sell in the organization. Always make sure you bring consumer DNA into the product. Your product should sell itself. But that does not mean you don't need sales people. So this is a really important distinction. Leverage everything about the internet, leverage everything about users to get to your customers. But you still will likely need sales as a way to help your customers navigate your product, help your customers navigate the competitive landscape and ecosystem. So you are going to want very domain specific sales associates that are going to be helpful for your customer in deploying enabled in these positions. But don't make that be a substitute, don't make that be a handicap for not building a great product. So you fundamentally build a product inside that. A company called Mixpanel comes in through the developer and eventually sells to that organization with a more inside sales process. Also read these three books: Crossing the Chasm, the Innovators Dilemma, and Behind the Cloud. These three combined, if you binge and read them all, you will come out ahead.

So in closing, today, right now is an amazing time to start a software company. I wish you the best of luck. If it doesn't work, we are hiring. The only other thing is, please do not compete with me because I have a lot of competition already. Ideally either come work with us or build your own company.

So thank you very much!

Lecture 13: How To Be A Great Founder

Reid Hoffman

As I looked through the syllabus of this class and thought what would be useful skills, what I’ve been thinking about is how do you see yourself as a founder? How do you think about what the skill set is? And what are the things you should be thinking about in terms of: am I ready? How do I get ready? Is it the right thing for me? These sorts of things.

So let's start with the perception of what a great founder is. And classically this tends to be Steve Jobs, Bill Gates, Elon Musk, Mark Zuckerberg, Jeff Bezos. And it's an image of founder as Superwoman, or Superman, who has this panopticon of skills. I can use the word panopticon because I am here at Stanford. It's things like, I can do product market fit. I am great at product, I am great at strategy, I am great at management. I can fundraise. I can do all of these skills and the thing you are looking for in a great founder, in the idea of the founder as a super person, is I am looking for someone who is awesome at all these things. They are well rounded, they are diverse. They can bat on all skills.

And part of how I found this emphasized, in the beginning of my own entrepreneurial journey, I remember reading an article that said Bill Gates is smarter than Einstein. And you are like, Bill Gates is really smart and is very accomplished, but I am not sure smarter than Einstein is really a phrase that even Bill Gates wants to be next to. It's partially because of this image of a founder as super person. Which is, a great founder can do anything. Jump over tall buildings in a single bound, all of these sort of things. And the reality is that a founder is someone who deals with all these different headaches and no one is universally powered.

Generally speaking you hope to have a couple super powers. Some things that are a unique edge to you, some things that are unique to the problem you are trying to solve, some things that will help you get an edge. Because competitive differentiation and competitive edge is super important. But it's not actually a function of genius.

And frequently it's very hard to tell the difference between madness and genius because usually it's the results that play out. Sometimes when dealing with uncertain environments you may even be genius and later may be thought to be a mad person. Or you may be a mad person and turn out lucky, you may turn out genius. It's actually a challenging set of how do you think about these sets of skills. And when us, mere mortals come into this sort of battle, what is the right way to think about it?

So when I thought about this question, on how is one a great founder, these are all skills that are super important. These are all things that you say, okay, this is really really important to do and you must in fact do this well. And it begins to look like a superhuman task. And what I did was decide to take a superset of these and focus on the interesting things to think about. What is it that actually makes a great founder? Because it's not actually that you score ten out of ten of these, you become the entrepreneurial olympian. You are actually the best at all these things.

So let's start with team. One way, I think, to explode the myth of super founder is usually it's best to have two or three people on a team rather than a solo founder. It's not to say that solo founders don't actually play out and they can, successfully. But most often two or three people is much better. When I look at these things as an investor, and I say what is a good composition of a project and founders that are likely to succeed, it's usually two or three of them. And the reasons are, we have already talked about that there's this very broad set of skills. There is this whole set of questions on how you adapt your company or be successful. If you have two or three founders, you have different skills you can compensate. Because, by the way, everyone has weaknesses. You can compensate for each others weaknesses. In the diversity of problems you encounter as a founder, you can actually attack them.

Other things I suggest when you look at, essentially, a founding team, is to have a real high preference for having co-founders, having a high degree of trust for those co-founders, because one other way on the whole entrepreneurial thing to die is you get a year down the road with your co-founders and then you are going through a messy divorce. That is not always but frequently fatal. And also the diversity of the tasks that you do. The next thing is location.

Frequently, I have had told to me, Oh Silicon Valley aggregates all this super talent, which it does. The reason why Silicon Valley startups are so successful is because all of these great people--immigration which is hugely important for talent and founders that immigrate here. Now if you think about it from basic math, even if you take something that Silicon Valley is super strong at, which is essentially software skills in the last two decades, not all the great software companies move here. Not all of them can move here. There are many of them in other various parts of the world. And so why do I put choice of location up there? It's one of the things it comes down to in thinking whether or not you are a great founder. Well the reason is, because of what great founders do is seek the networks that will be essential to their task. And they realize it's not just about, I am superman, I can do this anywhere. I can do this in Antarctica, etc, in order to be successful, I have to go to where the strongest networks are for the particular kind of thing that I am doing. And Silicon Valley, by the way, is super good at some kind of tasks, some places that you essentially try to solve certain types of problems. But it's not good at all of them. Let me take two examples.

One is Groupon, I don't think Groupon could have ever been founded here. Even though it's a software product, it even generates a network. Obviously a lot of the great networks are here and use internet technology as a mobile product and everything else. All of which we have a lot of great skill here in Silicon Valley and the networks are really good for this. One of the things that was central for Groupon for it's early days, was having massive sales forces. And a massive sales forces, strengthens and weakens a workforce, tends to go together. Silicon Valley, tends to be pretty adverse to plans that involve, Oh we are going to rent a twenty-five story building and in twenty of those stories we are going to have floors of sales people. And thats how we are going to get our thing going. That kind of plan here tends to not get a lot of interest, tends to get a lot of criticism, tends to not have talent aggregate to it, tends to have financiers talk about things like cap efficiency, and network effects, and other kinds of things that are key here. And so it's actually not a surprise that in fact, Groupon was required to be in Chicago, which is really good at this, as a way of getting going and showing that even software startups can be in other places. But even if you begin to think about it, you say what kind of other kinds of startups would someone be an idiot to move here to do.

Think of someone doing a fashion start up, not fashion a la Poshmark which is a mobile marketplace that are a bunch of things good here. But I'm designing a new fashion company. And I'm going to come to Silicon Valley to do it. That's actually not a great idea. The fashion company might be a great idea but you want the networks that support what you are doing. So part of the reasons why, where should I locate my start up, is a test for thinking about am I great founder is because part of the thing that happens when you are founding a company is, I will go to where this is successful to do. The metaphor I often use for entrepreneurship is jumping off a cliff while assembling an airplane on the way down. And the reason I do is because it's hard, it has a quasi mortal exit, which are default dead so you start taking every possible chance to actually win. So great founders will move to where the network is. This network is graphically Silicon Valley for tech startups, for mobile, for marketplaces this is a really good place to do it. For a bunch of other things, you should think about a different location.

Now here is something that's very en vogue. Very conventional to say you're a contrarian these days. So let's talk a little about what what a contrarian actually is. So it's actually pretty easy to become contrarian. It's hard to be contrarian and right. Particularly when you are thinking about, is my idea contrarian or contrarian enough. How does a smart person actually disagree with me? Because if you can't think of a smart person who isn't just ignorant or just crazy or anything else, but is a smart person that is somewhat of an expert may think that your idea has some serious challenges than it actually isn't contrarian. Contrarian is relevant to an audience. So when you are thinking about contrarian in terms of a really good contrarian idea is like, say it's consumer internet, okay what would other consumer internet people think is actually in fact not a good idea. And part of when you think about contrarian is to say, okay what do I know that others don't know? Because it isn't just, oh I'm brilliant and other people aren't, it's the reason my contrarian thing is right. It's a very bad test, that happened to be true, of course lightening can strike you in the field. So think a lot about what is it that I know that other people don't know. For example, in the very early days of LinkedIn, part of what I advise all founders to do is talk to every smart person who will talk to you and give you feedback. So with LinkedIn I walked around and said here's my idea, what do you think?

Two thirds or more of my network, including some very smart people, all thought I was nuts. The reason why they thought I was nuts was because I said it's a network product, it's only valuable with a bunch of people in it. The first person has no value until they invite the second one. Second person, first person have no value in it they already know each other. When do you actually begin to deliver on your use case, which is 500K to a million people. And so you are never going to get to size. It's never going to grow. Now what I knew was that the critics didn't know was that I could think of a set of different way by which people would say hey look I believe in the vision of this. I think it's interesting, or I think a product like this should exist, or I'm willing to play around with it. And I can level those sets of interest to grow the network to get to enough size that you can begin to deliver on the value propositions which Linkedin had. And that was the specific thing that I knew that the critics weren't thinking about. So when you think about being contrarian, you have to think about how is it that smart people disagree with me, that disagree with me from a position of intelligence. And there is something that I know that they don't know that will actually play out to be true. Now in this case, in general, as a founder it's good to be contrarian in the real sense.

Now the last part on the contrariness is to think about, there are lots of different ways to be contrarian. For example, a frequent one will be, others think you have a good small idea, but actually that's not small, it's large. Or actually in fact you can assemble the talent or while most consumer cellular startups tend to be, another LinkedIn example, only successful with the rocketships, actually a compounding curve can be very very valuable. LinkedIn never had it's rocketship moment, it was compounded year by year. But in consumer internet that becomes atypical in the pattern.

So here you begin to get to a bunch of sorts of problems that essentially founders run into. Well should I be doing the work? Or should I be recruiting people and delegating the work? And classically the answer to this is, actually in fact you need to do both. In fact, not only do you need to do both, you need to sometimes do one at 100% and sometimes the other at 100% and even though this is not so good at math, both at 100%. And so what you will see, this is sometimes classic, when you start thinking about what makes you a great founder, is you navigate what are apparent paradoxes. Another one I frequently talk about is, you've got to be both flexible and persistent. And the reason for this is entrepreneurs are frequently given the advice to have a vision, stay firm against your adversity. Realize that you have this vision that is contrarian to what people think and just stay on track. Get through the difficult times and get there. The other piece of advice given with each equal vigor is listen to data, listen to customers. Pivot, be flexible. Part of the thing this comes out to be in terms of being a great founder is to say well, when should I be persistent and when should I be flexible. And the vehicle I most often use for this is you should have a project you are doing, like a company, an investment, a thesis that essentially says why you think, possibly contrarian, why you think it is potentially a good idea. It should include what you know you think other people don't know.

And then as you are going into the battlefield, you go am I in fact increasing confidence in my investment thesis? Or decreasing value in my investment thesis? Because if I am increasing confidence then I hope to stay on track. Be persistent and, by the way, sometimes even with adversity your confidence can increase. If it's decreasing that doesn't mean jump out. Paypal, LinkedIn, Airbnb, a whole bunch of startups I've been a part of have had months where you are like, Oh my god why did we ever think this was a good idea? It's kind of a valley in the shadows moment. For example, at PayPal, it was August 2000, we were bringing in twelve million and the expense curve was exponentiating, we had no revenue, and a decrease in confidence. However we said, what do we do in order to fix that and that gives you your immediate action plan.

Another one is, should I have belief or should I have fear? Should you essentially go, well I have this vision of the way the world should be and I should ignore everything else and I should just go with that. Well again, part of what being a great founder is, is being both able to hold the belief, to think about where it is you want to be doing and want to be going, but also be smart enough that you are essentially listening to criticism, negative feedback, competitive entries. Where you are going, okay is this changing my investment thesis? Is this changing what I am planning on doing? It doesn't mean you lose confidence, you have the confidence but you also essentially have the patience. Again in this kind of thing is how do you put these two things together. Should I focus internally? Should I build a product, ignore the world, ignore competitors? Or should I focus externally, should I be recruiting? Should I be meeting people? Should I be gathering network intelligence? Again the answer is both. And the reason why I'm focusing on these type of habits, it's both rather than either or, is part of what makes a great founder is the ability to be flexible across these lines. To sometimes be 90% one way, sometimes be 80% the other way. Be executing the judgement on what does the current problem look like?

How is it that when I am trying to solve this that I should say this is what we should be doing and how should I be dividing the work? Part of when you think about these things is you say, this is another one that is classic, is people say well I am completely motivated by data its what customers say to user groups. I have a lot of entrepreneurial mythological, other kinds of things to talk about is gathering data, be guided on the data. Well actually in fact, data only exists in the framework of a vision that you are building, a hypothesis of where you are moving to. And the data can even be negative and you can think, well actually in fact this negative data means that I need to change or alter the way that I'm thinking about something. But I actually keep on a specific vision about what I'm doing. And by the way, sometimes when you have the specific vision you don't necessarily actually ever end up at that big vision that you were thinking about.

So for example, you know at PayPal we distributed these tshirts that said the new global world currency. Well actually in fact, I know Peter has been here, one of the jokes I told Peter was actually we do have this new world currency, what we are trading is in dollars you may have heard of it, it existed for a while. Where essentially a mass merchant for that-- now of course this is message is what's happening with Bitcoin though thats a whole nother topic there. However, the key thing is that vision saying we're creating a universal network that allows anyone to pay anyone to be a merchant to bring the electronics into the speed of commerce at any business that is being transacted. That vision kept a true north, but first we say well first we are going to have a banking model, then we think we are going to have a debt model, we are actually going to have a mass merchant model. How does that actually play out? So you are always combining the vision and the data, and data is within the framework of the vision. And sometimes the course of what you learn changes your vision. Now this is actually one of the ones that I we save this special picture for one of the ones that I actually think is quite key.

Normally entrepreneur founders are thought about as being the risk takers. Where everyone else cowers in fear at this notion of risk, they boldly go out. Now that's true, you have to be a risk taker, you have to be thinking about how do I make a really coherent risk because in fact the only really big opportunities, the only contrarian opportunities smart people disagree with you on happen to be ones that have more risks associated with them. On the other hand, part of the skill set, that when you are beginning to apply how you think about risks as an entrepreneur is how do I take intelligent risks? How do I take a focused risk, but if I'm right about that one thing then a bunch of other things break my way. And once I start doing that I try to figure out how to make my own shot possibility as high as possible? How do I minimize other risks? How do I essentially take this risk in an intelligent way that doesn't just go, oh yeah risk to the wind who cares but lets go. So this kind of combines that, this image is the best of the images that we found, is kind of the sense of that. Now back to what I was saying in terms of having an investment thesis.

Part of having a thesis is you chart it out as a list of bullets. For example with LinkedIn, everyone was going to be benefited by a public professional network, everyone will realize, including companies, that it's better to have it play out this way. The initial setup adoption will come from essentially people who visualize the world, play with it, and eventually the mass market will come on as they begin having a network, that is already having a network, with value proposition to them. That's what an investment thesis can look like and then you have economics, initially recruiting, and broadening those things.

You have that investment thesis, and you say is my investment thesis increasing or decreasing confidence? Do I think that the data that I get from the market, when I talk to smart people, how does that change my confidence in it? This is how you minimize risks. For example, very early days in Paypal, part of what happened was they said they were going to do cash and mobile phones with cash on Palm Pilots because its really easy. We actually realized the cash from Palm Pilots wouldn't work even before we launched the product. Basically what happened I went in and said to Max and Peter, I said here's our challenge--this group probably doesn't remember what Palm Pilots where, they were early PDA's. And so we lived in what was Palm Pilot central and the whole use case was splitting the dinner tab and everyone at table would have a Palm Pilots budget tab. Zero to one in every single restaurant. So you could, even by just thinking through the direction you are on you are going to hit a mine field and you need to pivot. And thats when Max Legend came up with the idea saying action packed sent by email. We can have email payments as the backbone of this and we were like yeah thats a good idea. Of course that what the whole thing pivoted into. And that is part of thinking though minimizing the risks as you are executing.

Here's another one that's kind of classic which is, should I have this long term vision or should I be solving a local near term problem? Again the answer is both these paradoxes. And the question is, you should jump between them. You should always have a long term vision in mind because if you actually completely lose your directions eventually you will find yourself somewhere in field thats not a good path out of. But if you are not focused on solving the problem thats immediately in front of you you're hosed. So part of the question about how to put these things together is you say, okay short term- what's the thing I need to be doing today? Have I made progress today? have I made progress this week? But is it largely on path? So I will give you an example of how this plays out in terms of financing or in terms of strategy.

People frequently think product strategy is fundamental to how startups- I have a product idea, thats a thing, I'm a founder. Actually the next level down on strategy is usually product distribution and whether its consumer internet or enterprise, or anything because actually in fact no matter how good your product is if it doesn't get the customers you're hosed.

So usually you have to have product distribution. It is more fundamental than what the actual product is. And the one below it is financing. The reason it's financing is because if you run out of money and the whole effort goes away, even if you have a really good idea, it doesn't work. So frequently when you are executing on a good strategy you are actually in fact, when I am raising money, this fundraising, I'm thinking about the next fundraiser. I'm thinking about how I'm set up for it, I'm establishing relationships that would be key to that. And I'm not executing like, oh the only thing that matters is I get to the next fundraising. You have this business that you are building, but I'm thinking that as a core strategy in terms of how I'm executing frequently you are thinking how does my product distribution work such that the financing works well. And that's kind of how you architect these things together.

So how do you know you might be a great founder? Well you should have some super powers. It's generally speaking useful to be a good product person. It's useful to have good skills about leadership, bringing networks in, persuading people, and it's useful to be able to- and this is kind of fundamental, is recognized whether you are on track or not. To have that kind of belief but also paranoid about am I tracking against my investment thesis? And when you do that the right way and you are learning and you are assembling people and you are assembling that around you. That's generally speaking how you end up being a great founder. Now classically, and I deliberately put up five white male pictures, is classically you have, these are the iconic founders. But in fact, founders can be very diverse. They can be extraordinarily talented at different areas because there's different kinds of external companies. There's different kinds of problems that they are trying to solve. And I don't mean diversity in term of gender, race, etc. Diversity in age, diversity in experience. Jack Ma was a teacher before he got into this. That's the kind of thing that you should think about.

So the question is how you cross uneven ground? How you assemble networks around you? How you get people to assemble? It's a constantly changing problem to face when you are trying to found a company. So I think the thing I was trying to get people to think about was, this is to say, there is not one skill set, there is an ability to learn and adapt. And an ability to constantly have a vision that's driving you but to be taking input from all sources and then to be creating networks all around you. And that's essentially what makes a great founder. So your ability to do that while crossing uneven ground in the fog, which is kind of the way entrepreneurs, did you always know this was going to work? No, unless you are crazy. Although sometimes crazy works. So with that I will now go to a few questions. But it was kind of this mindset of founders which is kind of key. And if theres' no questions, oh here.

Q: I'm curious how you targeted, you selected a different strategy to strengthen your investment thesis and help it take off. It seems like every startup faces that same challenge.

A: So one really fundamental thing is to think about product distribution as key. And for LinkedIn we had a couple things going for us. One, the web was boring in 2003. Basically what happened was everyone thought that consumer net was over. So people were doing clean tack and enterprise software and everything else. It's a much harder problem now. Because everyone thinks the internet and mobile is interesting. So breaking through the noise is really the key. So the strategy we used wouldn't work. We just basically set up, sent out some invitations to a group of people, and then tuned the mechanism, did PR. One of the people, one of the decisions we made early that was right was to say should we only allow it as invite only or should we allow cold signups? The reason we should allow cold signups is because the people who are super enthusiastic about this weren't necessarily the people we know said they would sign up and spread it. That sort of thing were all the kind of decisions we made. Now that challenge is much harder. The challenge when you think about product distribution is how are you competing for potential customers or potential members time? And what are they-what do they have to believe in? Back in 2003 was like oppression may not work that's potentially a good idea. What the hell? I'll play with it theres not a lot of other things for me to look at. Today theres tons of things. So your strategy today when you are looking at product distribution has to be what is my really decisive edge? What is the hack that I know that other people don't?

Q: How do you know if someone is a good founder or not?

A: I'm a huge believer in references. I only meet with someone when they come to me through a reference. So one of the things by the way is after this I have to run off because I have a meeting to get to. If you want to get time and attention with me, find a reference. That's not a pitch to using LinkedIn, it's a question of this is how you sort out time. Like Sam knows me, and so a reference to me is in fact the way I do this. So example, when I met with the AirBnB guys a part of the reason why I could interrupt them two minutes into their pitch and say, Im going to make you an offer to invest. I want to hear the rest of the pitch because I think what you are doing here is magical and awesome. Was because I had already had references on them. That was only two minutes, not thirty minutes. Because I had already knew about them before coming in. And by the way, by and large that is some version of what is true to most great investors. And its that network that's really key.

Q: Do you believe destiny of insight to be a great signal for great founders?

A: I would definitely say that the ability to say coherently what you are targeting, to articulate something that isn't trying to boil the ocean, or a swiss army knife approach. One focus, like you are right about this and it works. That is actually pretty important in being able to judge a founder. Because if you don't have that level of clarity, you are not going to be able to assemble the network behind you. You are not going to be able to get investors, you are not going to be able to get employees, you have to be able to articulate a very clear mission about what you are doing. Insight is helpful although a little bit of this depends on the stage. I find myself attracted to founders who have analyzed the problem in a good way. But often times I have seen great founders who have not present good analysis but have an instinct about what they are doing. So you more chart what is going on around them.

Q: How do you keep persistence when starting?

A: Well LinkedIn went through, for those of you who remember we were treated as the little alternative to Friendster, then Myspace, then Facebook. We had a lot of the, we are the little tiny one next to these respected giants each at the time. Ultimately, for me when I was thinking LinkedIn, this gets back to the investment thesis as a mechanism. I continued to believe the right economic system designed for every individual's life, and organizations life to have public professional profiles. But that world is the way the world should be. everyone is much better off with it. And we are getting closer to that than everyone else. It may be that it hasn't taken off as fast as I would have liked it. It may be that the general world has gone, oh the social stuff is really interesting. We could only get in the news the summer of 2003 by saying we were Friendster but for business. It's completely nonsensical once you begin to look at the thing. But it was like, okay we will cover you because you are Friendster but for business. But that was important to get people to begin to pay attention to us and so the confidence was, that world I still have confidence in, I still believe should exist and no one is getting closer to it than we. It's taking us maybe longer than I hoped to get there but thats okay.

Q: What is it that gets you wrong about someone who is going to do the distance?

A: To some degree you can only fully cross these minefields by going and doing it. So you can be wrong about your hypothesis. The kinds of things that frequently get you wrong or you think a person--- for example I frequently use in interaction is I push on the idea and what I'm looking for is both the flexibility and persistence. What I'm looking for is, I have conviction, and what I'm thinking I'm arguing it. But I am listening to what you are saying. And I am adapting to the concerns you have with that. Sometimes you will find someone that says, look I have learned to mimic that behavior so I have learned to say, for example I have learned to look like I am reasoning with you and I look like I am thinking about the challenges you bring up, but actually in fact I am ignoring you. Ignoring me might be fine. But ignoring the world in general is usually disaster.

And so these are the kind of things that in the measurement essentially getting wrong. Most often the kind of reference questions asked about founders is adaptability, one of the phrases I look for is infinite learning curve. Because each entrepreneurial pattern is to some degree unique and new. And can you learn the new one? Does the learning break down or is there some skill set? is there an ego issue that gets in the way? Like everyone must adulate me and that will cause you to behave wrongly in adapting to the problem. I think I have one last question.

Q: What makes a great co-founding team and what makes a good way to evaluate a co-founder?

A: The first thing is, it's super important to collaborate really well. That was the point I was making during the team. If in fact you don't have serious trust-- So the key thing is when you are thinking about founders is do you have a diversity of the necessary strengths across the board. Frequently you need one technical founder, at least. Frequently you need someone that is going to be dedicated to the business side for fundraising, these sort of things. A classically skill set and usually its some composition across them. Its kind of what you think of founders one, when you think about a founding team. When you get the next level deep, one of the things people classically tell you is don't invest in a husband and wife team. And that adds a little extra freight to it and everything else because the is extra dynamics and all that.

I actually think what you are looking for is, do they collaborate well. Do they help each other get to truth? So for example, I am most part when I am talking to a team. I am looking for when they are reasoning to each other, not just all singing from the same thing. But did you think about this? Or what about this as a challenge? You are navigating the field of battle which is a bunch of risks. For example, one of the things that was pretty common in Paypal is, Max who invented the prod systems and everything else, would frequently come into Peter's office, Peter Thiel, and say here are some things that are going to kill us and let me focus you on them. So it's not like we are all saying oh yes we are all singing our Kumbayas but we are adjusting to what is truth and what is the problem we need to solve? And what's the problem in short term and what's the problem in long term? How are we tackling it? And that collective problem solving, that collective learning is the kind of thing that actually makes a great teams.

Q: So different founders, different areas. How do you identify them?

A: The talk was aimed at what is unique about the mindset I think of founders. There is great founder across all. All founders-- there are differences. For example, in software speed to market, speed to learning is really key. In hardware if you screw it up you are dead. So accuracy really matters. If you build and ship the wrong thing you are hosed. So generally speaking as an investor, and this is part of the reason a lot of investors have a certain set of things they learn really well and reapply, is because they try to understand a domain really well to be able to identify which of the founders in this domain really matter and if we are investing in this domain how do we do that well? So there are attributes that are unique per domain. for example, one of the classic ones is, how good must you be at operational efficiencies in terms of margins, cost control, etc. You are dealing in the worlds of atoms including in commerce, you have to be really good at that. You are doing a digital game, like a Zing start up it doesn't matter at all, right? So you look for that sort of fit somewhere. Part of the beginning of this is that it's not in fact one person is good at at everything. One of the funniest conversations I had with a friend of mine who works for me at my first start up, Social Net. He looked at me and said, Reid I would never hire you to be a manager at McDonalds. I was, I wouldn't either. I would be terrible at that. So its the skill set that fits but also the whole point of this is actually being able to navigate a set of things that look like paradoxes. Sometimes being heavy on one, sometimes being heavy on the other. And having the right judgement at the moment at what you are doing and thats what tends to be more universal.

Q: How do you know when to pivot?

A: Part of the reason why having an investment thesis and your confidence in the rest of the place and being pretty clear on that is generally speaking the answer I give to people is if your confidence is unmeasured for a fairly long time or is decreasing, because measured for a long period of time it should be decreasing, then its decreasing and you go into intense mode. We try to figure out what types of things you can do to increase your confidence and thats failing. Thats a seriously good time to think about pivoting. You might have a thesis on raising money, you may have a thesis on what's the pattern on what the product distribution on growth or viral, or anything else will work. Well I tried these three things and this fourth thing doesn't seem as good as the other three. The next two things seem even worse. That begins to decrease your confidence. And that is when you should think about pivoting. A frequent mistake when it comes to pivoting is waiting until you have been crashed into the wall and everything is dead. And you can't maneuver any more. And thats you waited way too long. Now in times of personal career goals and so forth. One of the things I meant to talk to you during the slide, one of the classic questions is balance. I actually think founders have no balance. One of the funniest conversations I ever had was with the governor of Colorado, was like we are going to attract really great entrepreneurs here because we have this balanced lifestyle. Literally if i ever hear a founder talk about how I have a balanced life they are not committed to winning. So the really one great founders are, I am going to literally put everything into doing this. Now it may only be for a couple of years. I may do this for a while but hile I am doing this i am unbalanced. You are super focused on this. You work really hard and there are lots of ways to die.

Lecture 14: How to Operate

Keith Rabois

I'm going to talk about how to operate. I have watched some of the prior classes and I am going to assume you have hired a bunch of relentlessly resourceful people, you have built a product that at least some people love, hopefully raised some capital, and now you are trying to build a company. You have been forging a product and now you are forging a company. And I would actually argue, forging a company is a lot harder than forging a product. Basic reason is people are irrational. We all know this. Either your parents, your significant other, your brother or sister, your teacher, somebody in your life is irrational. Building a company is basically like taking all the irrational people you know putting them in one building, and then living with them twelve hours a day at least. It's very challenging. Now there are some techniques for coping with that some people get good at it, some people don't. But that's really what operating is all about.

So basically what you are doing when building a company is building an engine. At first you have a drawing on a white board and you are architecting it, and it looks especially clean, beautiful, and pretty. But when you actually start translating it to practice it actually starts looking more like this and you're holding it together with duct tape. It takes a lot of effort from people to hold it together, that's why people work 80-100 hours a week. It's that heroic effort to keep this thing together because you don't actually, yet, have polished metal in place. Eventually you want to construct a very high performance machine. A machine that almost nobody really has to worry about every hour, every minute. And as we used to joke about at eBay, that if the Martians took over eBay it would take 6 months for the world to notice. That's eventually what you want to get to.

As Warren Buffett says, build a company that idiots could run because eventually they will. So this is what you want. Basically a performance machine that idiots can run. Now as a leader, what is your real job, what's your role? Strictly speaking there is only one book ever written that actually explains how to do this. It's rather old, written in 1982 by Andy Grove, it's quite famous, and successful. And his definition of what your job is, is to maximize the output of the organization. Your organization that you are responsible for, so the CEO's (responsible for) everything and VP would be a part of the organization, and the organizations around you. So if you are a VPE, you are actually responsible for the performance of the product team and the marketing team because you have influence there. So this is how you measure people, and you want to focus on the output and not the input. The old adage about measuring motion and confusing progress. You are measuring only progress. And this is going to sound like a fancy and glamorous thing to do. Maybe people get excited about managing a whole large organization and being "responsible" for the output. But in practice, what you are going to hopefully learn today is that it's more about things like ordering smoothies, teaching your receptionist how to answer your phone properly, and serving as a $10 an hour TaskRabbit for your employees. So let's talk about that.

So at first, when you start a company everything is going to feel like a mess. And it really should. If you have too much process, too much predictability, you are probably not innovating fast enough and creatively enough. So it should feel like everyday there is a new problem and what you are doing is fundamentally triaging. So some things will look like a problem, and they are actually colds, they are just going to go away. So somebody is annoyed about this or that, that is a cold, you shouldn't stress about it and you certainly should not allocate all your time to it. And some things are going to present themselves as colds, but just like in the emergency room if they are not diagnosed properly they can actually become fatal. What I am going to try to do is give you frameworks for thinking about when things are colds and when things are potentially fatal.

So one of the most important things I learned at Square is the concept of editing. And this is the best metaphor I have ever seen in 14 years of running stuff, of how to think about your job. It's a natural metaphor, so it's easy to take with you everyday and it's easy to transmit to each of your employees so they can figure out if they are editing or writing. It's a natural construct, you generally know when somebody asks you to do something, am I writing or am I editing? So an editor is the best metaphor for your job. And we are going to talk about the specific things you are doing in editing.

The first thing an editor does and you have all probably had this experience in school, is you submit a paper, to a TA, a draft to your friend, and the first thing that editor does is they take out a red pen, or nowadays you go online, and they start striking things. Basically eliminating things, the biggest task of an editor is to simplify, simplify, simplify and that usually means omitting things. So that's your job too, is to clarify and simplify for everybody on your team. The more you simplify, the better people will perform. People can not understand and keep track of a long complicated set of initiatives. So you have to distill it down to one, two, or three things and use a framework they can repeat, they can repeat without thinking about, they can repeat to their friends, they can repeat at night.

Don't accept the excuse of complexity. A lot of people will tell you, this is too challenging, this is too complicated, yeah well I know other people simplify but that's not for me, this is a complicated business. They’re wrong. You can change the world in 140 characters. You can build the most important companies in history with a very simple to describe concept. You can market products in less than 50 characters. There is no reason why you can't build your company the same way. So force yourself to simply every initiative, every product, every marketing, everything you do. Basically take out that red and start eliminating stuff.

Second thing editors do, is they ask you clarifying questions. When you present a paper to someone, what do they usually do? They find some ambiguity somewhere and ask, do you really mean this? Did you really mean that? Give me an example of this? That's what your job is. So you are in a meeting, people are going to look to you. And the real thing you do, is you ask a lot of questions. And they can be simple basic questions like should we try this seven days a week? Or six days? They can be fundamental questions like, where’s our competitive advantage here? We try to do this as investors too. Some investors will ask you a billion questions about a billion things and they will have you do diligence forever. We try to narrow down to, what are the one two three four things that matter most to this company? And only focus on those things. So it allows us to be more decisive and we can make decisions rapidly. It allows us not to distract you from your day job which is actually building a company. And yet still I think we get to the highest fidelity question because we don't have all these extra extraneous details and data. Now it's hard, it's something you have to practice. But when you get good at it, every step you eliminate, Andy Grove estimated you can improve performance by 30-50%.

Now the next thing you do is you allocate resources. So the editor construct, this is what editors do all the time. They take editors from the Mideast, covering the Mideast and they move them to Silicon Valley, because Silicon Valley is more interesting. Or they move them to the sports section because they want to compete on the basis of sports journals and other publications. So that can be top down, where I take a whole bunch of resources and people and say, we are now going over here. We are going to compete on this basis. Then next month, next quarter, next year well that Middle East coverage is getting boring, we don't want to do that anymore. Let's go chase after something else. Or it can be bottom up, just like journalists mostly come up with their own stories.

The people who work with you, generally, should be coming up with their own initiatives. So a reporter, generally, who covers Google will come up with the interesting stories that they are hearing in ether and propose one or two to their editor for approval. But it's not the editor saying, go cover Google and this is the angle I want. Once in a while they do that, but its not the meat and potatoes of what a journalist does every day. Your goal over time is to use less red ink every day. So one way of measuring how well you are doing at communicating or talking to your colleagues about what's important and what's not, about why some things are important and why some things are not. It's how much red ink you are pulling out in a day, it's okay if you are having a bad day and the red ink is all over the place. But it's not okay if the red ink next month is more than it was last month, and next quarter more than this, so measure yourself by how much red ink you’re creating.

The other thing that is very important that actually isn't as intuitive to a lot of people, is the job of an editor is to ensure consistent voice. So if any of you read The Economist, you can tell that there is one consistent voice. You can pick up any article, any post in The Economist and it feels like it was written by the same person. Ideally, your company should feel, on your website, on PR releases, on your packaging if it's a physical product, anywhere on your recruiting pages it should feel like it was written by one person. That's extremely difficult to do. And at first you are going to be tempted to do that yourself, which is okay for a founder to do that him or herself initially. Over time you do not want to be doing all of the consistent voice editing by yourself. You want to train people so they can recognize differences in voice.

See this website page, it looks very different than this recruiting page. You start asking questions, why is that? Is the reporting messed up? Is one of the leaders over here not really understanding the voice of the company? You have to fix that over time, but you want to start with the objective that everything should feel the same. It's quite difficult to practice, almost every company has one piece of the organization that isn't on the same voice. At Apple, which is notorious, even under Steve's regime, which is notorious for getting this right, if you asked someone who worked at Apple, asked them about the internal tools about recruiting, do they really feel like Apple products? All of them will tell you no. So you are never 100%. But you definitely want to get as close to that as you can.

Next complicated topic is delegating. So just like the other metaphor on editing is writers do most of the work in the world, editors are not writing most of the content in any publication. So that is true of your company, you shouldn't be doing most of the work. And the way you get out of most of the work, is you delegate. Now the problem with delegating is that you are actually responsible for everything. The CEO, founder, there is no excuse. There is no, there is that department over there, this person over there screwed up. You are always responsible for every single thing, especially when things go wrong. So how do you both delegate but not abdicate? It's a pretty tricky challenge, both are sins. You over delegate and you abdicate, or you micromanage, those are both sins. So I'm going to give you a couple techniques for solving this.

First, and this actually came from High Output Management and Andy Grove, is called task relevant maturity. It's a fancy phrase for, has this person ever done this before? It's really simple, how mature is this person in doing something? And the more they have done the exact same task before, the more rope you are going to give them. And the more they are trying something new, the more you are going to instruct them and constantly monitor. This is a basic concept but it's worth keeping in the back of your brain. The interesting implication, and this is pretty radical, is that any executive, any CEO, should not have one management style. Your management style should be dictated by your employee. So with one particular person, you may be very much a micromanager because they are quite low on this scale. And with another person,, you may be delegating a lot because they are quite mature on this scale.

So it's actually a good thing if you do reference checks on somebody and half the people you call say they are a micromanager and the other half say they actually give me a lot of responsibility. That's a feature not a bug. I didn't understand that at first at all. I used to be befuddled when people would do reference checks on me and come back with this complicated mosaic. Then I finally figured out that maybe I was doing my job correctly. So then I taught others that this is the way to do it.

A more nuanced answer that I came up with, is how to make decisions. Delegating vs doing it yourself. You don't want to do it yourself too often. So I basically borrowed from Peter, this is my first two by two matrix ever in my life, but he taught me something at least. You basically sort your own level of conviction about a decision on a grate, extremely high or extremely low. There's times when you know something is a mistake and there's times when you wouldn't really do it that way but you have no idea whether it's the right or wrong answer. And then there is a consequence dimension. There are things that if you make the wrong decision are very catastrophic to your company and you will fail. There are things that are pretty low impact. At the end of the day they aren't really going to make a big difference, at least initially.

So what I basically believe is where there is low consequence and you have very low confidence in your own opinion, you should absolutely delegate. And delegate completely, let people make mistakes and learn. On the other side, obviously where the consequences are dramatic and you have extremely high conviction that you are right, you actually can't let your junior colleague make a mistake. You're ultimately responsible for that mistake and it's really important. You just can't allow that to happen. Now the best way to do that is to actually explain your thinking why. It's easy to shortcut when you get busy explaining ways in the world but it's very important to try.

When I was at LinkedIn, I had a colleague that was quite, quite talented but occasionally would get annoyed if I did not agree with his opinion on something. So I would spend a lot of time trying to persuade him why I was making a decision a certain way. And his wild card, his card he would call out if I didn't quite persuade him was, okay you're the boss. And that to me was like I was burning a lot of social capital. Every time he said that I knew I was creating a really thin line and ultimately that was going to backfire if I did that too often. You want to track the times that you are doing that.

An example of this is at Square, one of my favorite people in the world and my second hire, first marketing hire, had this program he wanted to run called Inner Square which allowed Square merchants to give out, imagine a food truck outside put out ten Squares on the counter and people could just grab them. And Kyle had this great idea that this would be an awesome marketing program. Squares would spread Squares to other people and to some extent it was on brand. So it didn't have catastrophic consequences. Each of these ten Squares didn't cost that much money, so financially we could afford to do it. But at that time, my ten years of experience said it was not going to work on a meaningful enough scale for our metrics and I preferred not to do it. Kyle was so excited about this that I decided to just let him do it. He learned that when you measure this thing, it's not massive. It doesn't create massive value for the company. It did require a fair amount of operational complexity to ship all these Squares to people and figure out how to get them, etc, etc. But it allowed him to be excited about his job and to learn how to filter future ideas. So it was totally worth letting him make the “Mistake."

The next and most important thing you do is edit the team. So these are the people you work with. Nobody is going to have a perfect team and you certainly aren't going to start that way. So what I am going to try to do is maximize the probability of success in editing the team. So I like this idea of barrels and ammunition. Most companies, when they get into hiring mode, as Sam pointed out you should defer that a bit, but when you do just hire a lot of people, you expect that when you add more people your horsepower or your velocity of shipping things is going to increase. Turns out it doesn't work that way. When you hire more engineers you don't get that much more done. You actually sometimes get less done. You hire more designers, you definitely don't get more done, you get less done in a day.

The reason why is because most great people actually are ammunition. But what you need in your company are barrels. And you can only shoot through the unique barrels that you have. That's how the velocity of your company improves is having barrels. Then you stock them with ammunition, then you can do a lot. You go from one barrel company, which is mostly how you start, to a two barrel company and you get twice as many things done in a day, per week, per quarter. If you go to three barrels, great. If you go to four barrels, awesome. Barrels are very difficult to find. But when you have them, give them lots of equity. Promote them, take them to dinner every week, because they are virtually irreplaceable because they are also very culturally specific. So a barrel at one company may not be a barrel at another company.

One of the ways, the definition of a barrel is, they can take an idea from conception and take it all the way to shipping and bring people with them. And that's a very cultural skill set. Two questions are probably occurring to you. How can you tell who is a barrel and who is not? One is you start with a very small set of responsibilities, it can be very trivial. It can be something like, I want to reward the engineers in my office at nine o'clock every night with a nice cold, fresh smoothie. This is actually a real example. I was frustrated, our engineers were working really hard, and maybe 20%, 30% would stay late in the evening and we had already served them dinner but I wanted to give them something cool to reward them. You can think about alcohol but that's a little complicated. So smoothies were probably a little bit better than pizza, which drains you of energy. But nobody could get smoothies to show up in my office at nine o'clock sharp, that were cold, that tasted good, and that were delivered in the right place that the engineers would find them.

You would think this is simple but in fact it took two months to get this done. So we had an intern start, and I think on his second day I was explaining this problem, and he said, well I will do it. And I was looking at him like there was no way. I have seen my office manager fail, my assistant fail, who were actually pretty good. This just isn't going to happen. And low and behold they show up. On time, cold, delivered at the right place, and my first instinct was great. Nothing about the smoothies, but now I can actually give him something more important that is more complicated to do.

And that's actually what you want to do with every since employee, every single day, is expand the scope of responsibilities until it breaks. And it will break, everybody, I couldn't run the world, everybody has some level of complexity that they can handle. And what you want to do is keep expanding it until you see where it breaks and that's the role they should stay in. That level of sophistication. But some people will surprise you. There will be some people that you do not expect. With different backgrounds, without a lot of experience that can just handle enormously complicated tasks. So keep testing that and pushing the envelope. The other signal to look for is once you've hired someone, with an open office, just watch who goes up to other people's desks. Particularly people they don't report to. If someone keeps going to some individual employees desk and they don't report them, it's a sign that they believe that person can help them. So if you see that consistently, those are your barrels. Just promote them, give them as much opportunity as you can.

The other question everybody asks about people is when do you hire somebody above somebody. And when do you mentor somebody, and when do you replace somebody? And the way to think about this is that every company has their own growth rate, and every individual has their own growth rate. So some companies that are very successful, lets say

LinkedIn. LinkedIn was always a very linear company, it never went like this. So for example I joined LInkedIn 18 months after we launched and we only had 1.5 million users. Which for a social product is a very small number. And when I joined I was the twenty-seventh employee, and when I left two and a half years later they only had fifty-seven employees. In contrast, when I joined Square as the twentieth employee, two and a half years later we had two hundred fifty-three employees.

So each company has it's own velocity on this curve. So if the company is going like this, you can only keep people on the roles if their own learning curve is going like this. On the other hand, if your learning curve is like this, anyone learning faster than that, you can give them the same roles as they do. So always track the individual slope of employee and the company growth rate. Now that you have your barrels figured out, and you can identify people who can take ideas that you have in the back of your head, scope it out, run with it, ship it, and it's perfect. Where do you aim these barrels?

So I am going to argue that you need to spend a lot of time focusing on people. This is something I learned from Peter Thiel actually. He used to insist at PayPal that every single person could only do exactly one thing. And we all rebelled, every single person in the company rebelled to this idea. Because it's so unnatural, it's so different than other companies where people wanted to do multiple things, especially as you get more senior, you definitely want to do more things and you feel insulted to be asked to do just one thing.

Peter would enforce this pretty strictly. He would say, I will not talk to you about anything else besides than this one thing I assigned you. I don't want to hear about how great you are doing over here, just shut up, and Peter would run away. And then focus until you conquer this one problem. And the insight behind this is that most people will solve problems that they understand how to solve. Roughly speaking, they will solve B+ problems instead of A+ problems. A+ problems are high impact problems for your company but they are difficult. You don't wake up in the morning with a solution, so you tend to procrastinate them. So imagine you wake up in the morning and create a list of things to do today, there's usually the A+ one on the top of the list, but you never get around to it. And so you solve the second and third. Then you have a company of over a hundred people so it cascades. You have a company that is always solving B+ things which does mean you grow, which does mean you add value, but you never really create that breakthrough idea. No one is spending 100% of their time banging their head against the wall every day until they solve it. So I highly recommend some version of that. You can be less stringent, you can give people three things to work on, but I would still track the concept of what would happen if you only gave everybody one thing to prioritize.

You don't want to be making all these decisions yourself. You have to create tools that enable people to make decisions at the same level you would make them yourself. So how do you create scale and leverage? The first thing I would recommend is to build a dashboard. This is an old Square dashboard, it looks pretty presentable even today. The construct of the dashboard should be drafted by the founder. You need to simplify the value proposition in the company's metrics for success on a whiteboard. You can have other people build the dashboard, I don't care about that. But you need to draw it out. Like what does business success look to us and key inputs to those and then have someone create something that is very intuitive for every single person in the company, including customer support to use. And then, the key metric of whether you succeed is what fraction of your employees use that dashboard every day? If it's actually useful, it should be close to 100%. It's not going to be probably 100% but you want to measure that. Just like you have quality scores for your other KPI’s with users, your dashboard needs to be as intuitive as your product is for users.

Another concept is transparency. Transparency people talk a lot about, it's a goal everybody ascribes to but when push comes to shove, very few people actually adhere to it. So let me walk through a little bit of transparency and different stages of transparency. Metrics are the first step. So everyone in your company should have access to what's going on. Other things I like to do, is to take your board decks. As you get more formal, the board decks will get more complicated. And actually review every single slide with every single employee after the board meeting. You can strip out the compensation information if you really want to. But every other slide you should go through with every single employee and explain it. If you can remember some of the feedback you got from your board that is really cool to pass on.

Another thing we did at Square is when we scaled, not everybody is going to get invited to every meeting, but they are going to want to go to every meeting. The way you scale that is you create notes for every meeting and you send them to the entire company. So we created notes at least for every single meeting that involved more than two people, somebody would write notes and send it to the entire company. So everyone felt that as the company added employees, they continued to track what was interesting, what was going on. And they never felt excluded, hopefully. Another thing is, even details around conference rooms. Every conference room at Square has glass walls. Because as soon as you have regular walls, people wonder what's going on. It's amazing, if they can see who exactly is in the meeting and who is meeting with who when, they don't start to worry nearly as much as about what's going on behind those closed doors.

Stripe, you may have seen a blog post about, I think Patrick wrote it, about email transparency, about actually allowing everyone to have access to email. That's pretty far out there but it has certain merits to it. I would call all the tactics you hear and read about as minimal viable transparency. I actually think you could push the envelope a lot more. Steve Jobs actually tried this at Next, he actually tried transparent compensation. I actually think that even though Next didn't do extremely well, the real reason wasn't around experimenting with compensation in transparency. There is a lot of merit to that. The critique of compensation transparency is, well we want people to be teammates and work together and collaborate. And if you look in the sports world where people are teammates and collaborate, all of their compensation is actually public. In fact, any one of us can look up one of their compensations in the sports world to get it exactly accurate. And somehow it seems to work. So I am not completely bought into that you need to keep compensation non-transparent.

And finally, metrics. So you want to measure things. You want to measure outputs, not inputs. And again, you should dictate this yourself. You should draft the dashboard yourself to tie this all together. One important concept is pairing indicators. Which is, if you measure one thing and only one thing, the company tends not to optimize to that. And often at the expense of something that is important. Cost is example of payments and financial services is risk. It's really easy to give the risk team the objective and say, we want to lower our fraud rate. It sounds great. Until they start treating every user in this audience as a suspect because they want to lower the fraud rate. So they require each of you to call them up on the phone and give them more supplemental information and fax in things. Then you have the lowest fraud rate in the world, you also have the lowest level of customer satisfaction score.

What you want to measure at the same rate as your fraud rate, is your false positive rate. That forces the team to actually innovate. Similarly, you can give recruiters metrics around hiring. And guess what? You will have a lot of people come in for interviews. But if you are not tracking the quality of interviewers, you may be very unhappy about the quality of people you are hiring and giving interviews to. So you always want to create the opposite and measure both. And the people responsible for that team need to be measured on both.

Finally around metrics. One insight I have had over my career is what you, you kind of want to look for the anomalies. You don't actually want to look for the expected behavior. So a famous example was at PayPal. None of the top ten markets that the company was planning on going after included eBay. One day, someone noticed that 54 of the sellers actually handwritten into their eBay listings, please pay me with Paypal and brought this to the attention of the executive team at the time. The first reaction from the executive team was, what the hell is going on? Let's get them out of the system, that is not the focus. Fortunately, David Sacks came back the next day and said, I think we found our market. Let's actually build tools for these power sellers instead of forcing them to write into their listing, pay me with PayPal. Why don't we just have an HTML button that they can just insert? And that actually worked. Then he thought, why should we have them insert it each time? Why don't we just automatically insert it for them? They can just insert it once, then every listing they have forever will have it automagically appear there. So that became the success for PayPal. Similarly I was at LinkedIn and I saw this stat that made no sense to me. The UI of the site was a little different then. 25% of all clicks, maybe 35% of all clicks from the homepage were people going to their own profile. And that didn't make any sense whatsoever. It was in the settings, you had to go to the margin and find a link. It was 25-35% of every click at scale, so this is just invalid stuff. And it made no sense whatsoever. I had never seen UI perform that way.

I went around for weeks trying to figure this out, then someone smart, actually it was Max Levchin, said something to me and I was like, he was like, it's vanity. I was like, ah ha! People are looking at themselves in the mirror. Thats a very good answer, because they weren't editing their profile. Nobody has something to edit everyday in their profile. But they were just looking at themselves in the mirror every day because it made them feel good. Then you could test that with, if I had more content would I look at myself in the mirror more often? It turns out, you did. If you had more endorsements would you look at yourself more in the mirror? You did. So we figured out what was underneath the utilitarian product, the product the team thought they were building was actually more emotional vanity. It didn't actually translate to the best feature like the Paypal example. We couldn't exactly put a button up that said, be more vain today on the homepage. That probably would not work perfectly. So it really never really took off like the Paypal example did. But it really clarified what users of the product really wanted. And we wouldn't have found that if we hadn't looked through anomalous data.

The final topic I want to talk about is details. And in my assigned reading there is a great book by Bill Walsh, called The Score Takes Care of Itself. And the basic point of the book is that if you get all the details right, you don't worry about how to build a billion dollar business, you don't worry about how to have a billion dollars in revenue, you don't worry about having a billion users. Thats a byproduct of what you do everyday to get the details excellent. So the topics that he talks about in the book that really resonated with me was, he took over the 49ers in 1979. They were the worse team in football, I believe they were 2 in 14 which is really bad if you don't know football. In the next ten years he transformed the team into NFL's best, won three super bowls. And what's the first thing he did to go from the worse team to one of the best in many ways? He actually taught the receptionist to answer the phone properly. He wrote a three page memo on how to answer the phone.

And that may sound absurd but what his point was organization as a whole does everything exactly the right way. Then receivers will start running their routes at 7 yards not 8 yards. And that actually will matter. And if every person on the team executes to the same level of performance, you will have a team that is performing at the highest possible level. And at the highest performance level, the team will play at their best. So how to relate this to a company may include a lot of details that do not matter, not seem that they matter superficially. Most people would agree that details matter when faced with the user. But what the real debate it is on things that don't face the use. Steve Jobs famously in the Mac, insisted upon an immaculate circuit board, you can read about this in various books. The Mac, for those of you who don't remember the Mac. Maybe most of you here, but may have seen it. It couldn't be opened. So the circuit board couldn't be seen by any person in the world. There was no way to open the Mac except by the people that worked at Apple.

Steve insisted that it be absolutely perfect and beautiful. That is the sort of detail obsession that building this sort of company requires. Examples that may be a bit more practical for you instead of circuit boards may be, what sort of foods do you serve people? It actually matters more than you might guess. When people don't like the food you serve them, what do they do? They go gossip, they go complain to their friends, they walk over to someones desk. Then all of a sudden that lunch that they are complaining about is what they are spending most of their time gossiping and complaining instead of brainstorming. You don't have this serendipities idea matching another serendipities idea that creates a spark instead they are all wallowing and whittling around. The best thing you can do is give people the food they want or the food thats good for them that makes them more productive. So it may seem like this glorious job you thought you had is more like running around being a TaskRabbit for people. But it is to take things off their plate that is a distraction so they can be high performance machines. And if you take enough things away from people to distract them and give them the tools to be successful, all of a sudden your organization produces a lot more.

Similarly, often one that people get wrong is office space. So one natural thing is when you need an office to have an office manager of your team go out and find offices. And they will go out and come back with photos and ideas. You need to do that yourself. The office environment that people work in everyday dictates the culture that you are going to be in. And the final thing, then I am going to take some questions, is around effort.

Ultimately I don't believe you can create a company without a lot of effort and that you need to lead by example. So Bill Walsh, in the first chapter of his book, he asks this question, how do you know you are doing the job? And this is the quote that he gave everyone when asked that question. So if this is how you feel everyday then you're probably on the right track. If it doesn't sound appetizing then you shouldn't start a business truthfully. And with that I am done with the prepared part and I will see if anyone has any questions that I can try to be helpful with.

Q: So you talked about making compensation transparent. How would you do that, especially when people equate themselves to the value of how much their salary is?

A: I would do it probably in bands. You can do it, just everybody in the company gets paid the same. Or you could have all discipline, all engineers-- Or you could do it by experience. How Steve did it at Next was there was a high band and a low band. You either had a lot of experience or a low experience and that was it. So low band, now would be about $85,000 flat. Everyone flatly gets paid $85,000, if you are a supervisor with experience everybody gets paid $130,000 and that's just it. Sort of the Next translated for inflation.

Q: So the question is, besides food, what sort of details do others care about?

A: The laptops they use. This is just a default everybody has. Five years ago it was a benefit to get people high powered machines as opposed to optimizing our cost by having Dell Machines and ugly monitors, just as an example. If you think about all these people who are relentlessly resourceful and equally talented in a mass competitive ecosystem competing for talent, you learn to give people the best possible tools to do the best possible job. So rigorously asking, how do I make people more successful, what things do they not need to be working on, or distracting. And what things can I give them to make them more valuable per day? Then just break that down per day and solve that stuff yourself.

Q: When you are in a startup environment, resources are scarce. How do you optimize for those things?

A: That's a good question. First of all I think you must have your own office. I don't believe ever in shared office spaces. Peter talks a little about this, that every good startup is a cult. And it's really hard to create a cult if you are sharing space with people. Because a cult means you think you are better than every other startup, you have a special way of doing things that's better than anyone else in the world. And if you are sharing physical space with others it's very hard to internalize that. So I would start there. But it is a prioritization question. Everywhere is scarce, so it's just a question of the magnitude of zeros you are paying attention to. Probably not $10 expenditure, but $100, $1000, $10,000, then $1 million starts being a rounding error.

So I would figure out is what is the most important, a quality office that creates a good vibe that allows you to recruit people. Because recruits are very savvy about this. They walk into your office and they can tell a lot about the culture instantly. I can walk into a company office and I can tell whether or not I am going to invest as soon as I walk in. Seriously, I just absolutely rule things out that I don't want to invest in as soon as I walk in. And there's times I walk in an office, like wow this is really impressive. You can tell how people work together, how hard they are working, how distracted they are. Roelof Botha at Sequoia made a point to me about YouTube. So when I invested in YouTube in the very very beginning, it wasn't obviously going to be successful. Then Roelof led this series A Investment for this client on YouTube and we were at a board meeting together and he said, I really think Youtube is going to work. I said, Why? And he said, every time I go to one of my portfolios companies half the office is watching Youtube during their lunch. I was like, really good sign. And so you pick up on these little things and you can predict a lot.

A: What do you think is the best way of getting street credit as a new manager?

Q: Oh boy. Almost all good managers in Silicon Valley are promoted because of their individual performance. In cultures that are bureaucratic the percentage is even higher. So we tried in PayPal to only promote people who were kicking ass at their discipline. Peter didn't believe in general managers. In fact I remember going for a job around campus with him my first week of PayPal. He was asking me how things were going, other kinds of CEO questions. Then we got in this debate about whether the company needed any more managers. He was like, nope. No managers. We are only going to promote people, so the VP in engineering is going to be the single best engineer. The VP of design is going to be the single best designer. The VP of product, is going to be the single best product person. And they are going to learn to manage later. The advantage of that is you don't demoralize people. Because everyone knows their boss is actually better at their job than they are. And they can learn stuff. And you can learn a little bit of the management techniques later, as opposed to promoting people who are just good people managers that don't have the discipline and skill.

That demoralizes people, so I think just being excellent at something and then being excellent at getting a bunch of people to do something is the next task. But people, some things you just have to learn by doing. Some people can’t learn to play the guitar by reading a book. You have to actually try to manage a bit and you won't do well at first. I have another set of tactics and classes on what you actually do when you transition from an individual to a manager. It's hard. One of the first things people don't get right right away is their timing allocation. Actually, what I would recommend is doing what I call a calendar audit. And track what you spend your time on in a month. How much is editing, how much is writing, etc. And optimize it over time. You can get a mentor. Find someone who’s been a manager before that will work with you. Not your boss, because your boss has a set of complicated objectives including how much are we shipping. A mentor can just focus on you and making you more successful.

Q: What are some things you can do to ensure a consistent voice in the company?

A: So I would look at every piece of copy in every department. Another part that is hardly ever, look at your recruiting website. It almost never has the same quality as your conversion funnel. I would look at customer support. Another classic area that is never up to the same quality. Treat customer support as a product so you actually have an engineering team and a design team that over time focuses on making that world class. Usually we have other executives at a scaled company. Most executives were trained differently at other companies and bring that with them. You have to crosstrain that. So if you hired a VP of engineering of Google its very different than a Design Leader from Apple. They don't actually learn anything the same way. So you are going to have to stitch that together some how. Either one or the other is going to have to learn the other style. Or you are going to have to create your own style and really teach that to your executives. So it shows up all the time. All you have to do is pick up the company's products and look for things that have a different voice and you can see it, visual voice, word choice, all over the map.

Sam Altman: Can you talk a little about the tactics of how you manage people? How often do you meet with them?

A: So the canonical advice sounds obvious but was radical back in 1982 when Andy wrote his book, is to have a one on one roughly every two weeks. Some people say every week, but I wouldn't go longer than two weeks. Every week can be ideal in many companies. The reason why there is another adage, you should only have five to seven direct reports. That actually derives on the concept of having a one on one every week. The director reports is so you can fit enough one on ones in your calendar a week and get other things done. I think one on ones once a week is ideal. I think the agenda should be crafted by the employee that reports the manager, not the manager. The one on one is mostly to benefit the employee. They should walk in with, these are the three things I want to talk about. Even bullet points in advance by email so you have time to chew on it, and you're not on the fly winging your answers. But that's probably the best structure. But if someone is really good and been doing this a long time and has internal credibility you might push out the one every week to once every two weeks. Maybe every month. I don't know that I would go beyond once a month ever.

Q: When is it acceptable to compromise and hire someone that is ammunition instead of someone who is barrel?

A: Truthfully you are going to hire more ammunition, naturally, than barrels. So there is a ratio between the two. The question is the ratio. At some point the ratio is going to get out of whack, you will have only one barrel in the company and fifty engineers. You might as well have ten engineers because you are not going to get any work done. You are just wasting resources. You are going to frustrate engineers because everyone is going to need your approval. Your signoff. Your editing. It's just going to stack and get frustrating to people. I think roughly one, to ten, to twenty is the right range. You don't need more engineers until you have more barrels. Designers are a little different. But you are always going to be hiring more ammunition than barrels. A good barrel will have a feel for that. One way to correct for this is, there is this natural tendency to create headcount on your team. Like an empire building tendency. Like I manage twenty people and Sam only manages ten and you manage three. So I'm more important than Sam. And Sam is more important than you. Put an X here, over Y which equals the output, and specify how many things they have done successfully and then divide by the number of people on their team. And tell them this is going to be the grade for their performance review. Shockingly the Y doesn't start increasing on that team. It's amazing how this works and be explicit about it.

Q: How often, as a venture capitalist, do you meet and interact with your companies?

A: When we invest, we do do some seed companies where we invest less money, but when we invest in a series A or B round, we join the board and roughly I meet with the founder or CEO every two weeks. That's the default. Obviously there are inflection moments when things go right or wrong and then it's on an ad hoc basis. Actually right now I do a lot by text message. I even have one CEO who Snapchats me all the time. Which I would actually rather he not, but the world has changed a lot. But I try in person meetings every two weeks.

Q: ?

A: No. Being a venture capitalist to me is like being more of a psychologist. So if you come to my office we have two chairs with a table in the middle. And we sit down and it's like, Tell me your problems. My response is usually, have you thought about this? Have you talked to this person? Have you tried this? Etc. It's just asking a lot of questions and going back like that. But that's 90% of what I do.

Q: ?

A: Well it depends on where your prioritizations are. Sam talked a little about this in his lecture. So a company will move recruiting first, second, or third. Somewhere in that spectrum. If it's your number one priority then 25% is probably a good allocation. Actually I like the calendar audit more for CEOs even moreso than new managers. So when I work with CEOs that aren't thriving in that role for the first time, I actually force them to show my their calendar. Now I am going to ruin this trick. But I ask them to write them out on paper and specify whatever they are. Then I ask them to pull out their calendar and see if it matches. And it never matches. Never. Recruiting is the one that is usually the most often awry. Half the CEOs will say recruiting is their number one priority. It's almost never the biggest block of time on anyone's calendar. So that is what you are trying to do is match resources to priorities in the calendar audit. And there is no software that does this really well for you. It would be great, right now we pull up someone's Google calendar and hand count up the hours. Which is insane. But that is the best way, just ask about their priorities. Priorities are raising money, you don't want to allocate most of your time recruiting. One more question.

Q: On the surface some of this advice is contradictory. How do you harmonize those roles?

A: It's a good question. How do you harmonize when details really matter, but you've got that one thing to do, you want to only allocate time to those one, two, or three things. How do you put those things together? There is some tension even in a healthy organization. There is some tension of why are we focused on writing the script as opposed to something the user may see.

The underlying philosophy of getting the details right is pretty important to install in the very very beginning of a company. Because people will start acting that way and making decisions that way themselves. So you won't have to literally do that. If you have to actually do that then that shows the foundation isn't that solid. When you first start the company it's about getting the details right. Everyone is precise, everyone on every task is always thinking that way. And then that scales, the people you bring in will think that way, the people that think that way will tend to get hired and those who can’t wont get hired. Each team and each leader will tend to enforce that themselves. So the CEO is almost never doing it. So it's partially how do you start. And the key to culture is it's a framework for making decisions. And if it's baked into your culture, people learn how to make decisions across that culture without you ever saying anything. You never have to really do anything except watch and promote and move people around.

Cool. Well I guess that's it.

Thank you.

Lecture 15: How to Manage

B Horowitz

When Sam originally sent an email for me to do this course, he said "Ben can you teach a fifty minute course on management?" And I immediately thought to myself, "Wow, I just wrote a three hundred page book on management. So that book was entirely too long."

I didn't actually have time to collapse the three hundred pages into fifty minutes. Like Mark Twain, I didn't have time to write a good short letter, so I'm going to write a long letter. But in this case, I am going to teach exactly one management concept.

I see CEO's mess up this one management concept more consistently than anything else. From when they're very, very early to when they're very, very big as a company. It's the easiest thing to say and the most difficult to master. The concept in musical form is from Sly and the Family Stone. "Sometimes I'm right and I can be wrong. My beliefs are in my song. No difference what group I'm in."

That's the musical version of today's lesson. For those of you who are musical, you can leave now.

When you're making a critical decision, you have to understand how it's going to be interpreted from all points of view. Not just your point of view and not just the person you're talking to but the people who aren't in the room, everybody else. In other words, you have to be able, when making critical decisions, to see the decision through the eyes of the company as a whole. You have to add up every employee's view and then incorporate that into your own view. Otherwise your management decisions are going to have weird side effects and potentially dangerous consequences. It's a hard thing to do because at the point when you are making a decision, you're often under a great deal of pressure.

Let's get into the agenda. I am going to cover four cases. First, I am going to cover demotions, which are very emotional. Then raises, which are also emotional. Then we are going to evaluate one of Sam's blog posts, which is news to Sam. I figured I'd tease him since he invited me to do a fifty minute management class after I wrote a three hundred page book.

Then I'm going to talk about history's greatest practitioner at this, I'm wearing a shirt with him on it, and how he used it to do something that nobody had ever done and has ever done since in human history. Basically complete mastery of the technique I am going to talk about.

So, first business example, you've got an executive, and do you demote or do you fire him? This comes from an actual conversation, an actual real life situation that I was working on with a CEO. The basic situation was this: he had a great executive he had hired. He was working harder than anybody else in the company and was doing everything he was supposed to. Everybody liked him because he worked so hard and was a generally a smart person, but he was in over his head knowledge-wise. He did not have the knowledge and the skills to do what the company needed him to do or really compete against the competition. So he couldn't actually keep him in the job, but he was a great guy. So the question is, should I fire this person or can I just move him into a lower role and bring in a person above him. That would be cool. Let's look at how you make that decision.

You are, in this case, the CEO. It's really hard if somebody comes to work every day at six AM, is working until ten PM, and is working harder than anybody in the company. It's really hard to just say, "Well sorry, nice effort but you don't get an A for an effort. You get an F because I fired you."

Nobody wants to have that conversation. A demotion is kind of neat because from a CEO's point of view, he can keep him in the company. He works so hard. He's a great example of somebody who gives great effort. He's got a lot of friends in the company, so from a cultural stand point it's a win/win because he gets to stay. Then I can bring in somebody who can solve my problem but I don't have to create another problem.

If you think about it from the executive's perspective, it's like, "I don't want to be demoted but I really don't want to be fired because if I get fired, that's a way harder more complicated thing to explain to my next employer that I got demoted. Getting demoted is, well I didn't really get demoted. I got a new job, a smaller title."

The last thing it enables, theoretically: the company values all of our employees. We brought you in. We made a commitment to you as an employee and it will enable you to keep growing with the company. This was the conversation I was having with the CEO and I said, “Well, wait a minute. Let me ask you this. What's the equity package that this executive has?" He goes, “Well, what do you mean?" I was like, “Well I would like to see direct level of compensation? Does he get a Vice President level of compensation? Does he have 1.5%? Does he have 0.4%?"

That gave the CEO pause. He's like, “Well, he does have 1.5% of the fully-diluted equity of the company." And I was like, “Ok. So you're an engineer in your company. How do you feel about somebody who used to be the head of sales who brought in with 1.5%? What do your engineers get? Do they get 0.1% of the company? 0.2%? What are they getting at this point? How are they going to feel about somebody who is NOT the Head of Sales with one point five percent of the company?" And he was like, " Uh oh." And I was like, “Yeah! Uh oh. Because how fair is that? Are you going to take the equity away? Are you up to do that? Are you up to go back and take back his compensation? How productive do you think he'll be if you take away his compensation? Secondly, will people give him the same respect now that you've demoted him? Because they knew him as this and now he's that. "I knew you when you were Head of Sales now you're the Regional Manager and you're telling me what the f* to do? You're telling me I need to make that call? It seems to me you got demoted. Who are you to talk to me? I am the up and comer. I am going to be the next VP of Sales at the company."

All these things come into play. When you look at the end, you may think you are dealing with one person. You may think that this is a demotion or a firing of one person. What does it mean according to that one person? But what you are really doing is saying, what does it mean to fail on the job? Particularly the highest paid, the highest compensated job in the company from an equity standpoint. And then, what's required to maintain your equity? Is it good enough to put in an effort or do you have to get a result? In different situations at different levels, the answers will be different. If this had been a person who was not an Executive brought in from the outside, but someone you promoted past where they should have been and didn't ever get that equity, maybe you make a different decision but you have to understand what it's going to mean to everybody, not just the person you're talking to.

Example two: An excellent employee asks for a raise. A good employee, this isn't like the last employee. First thing you think is they're really good, they asked me for a raise, they didn't ask me for no reason. They asked me because they think they deserve it. I want to retain them. I want to be fair! They've done a great job. I know that if I give them this raise, it's going to be all love coming my way. If I give you a raise we're good. You got a raise! It's awesome.

From your perspective you know what you want to do when somebody asks you for a raise. What about from their perspective? How would they take it if you gave them the raise. You have to remember, for them to get to point where they ask you for a raise, they did not wake up one morning and say, "I am going in and asking him for a raise." This is something they've thought about a lot. They've compared their other options. They may have an offer from another company. It’s something their spouse probably has been talking to them about. It's a serious thing. If you give it to them, they're very likely to feel very good about it. They may be paranoid, like " Why you giving me a raise?" But very unlikely. Much more they'll feel like...

(Plays "The Shmoney Dance" video)

For those who don't know, that's Bobby Shmurda and Rowdy Rebel doing the Shmoney Dance. That's the reaction you'll get. So there is a lot of momentum to say, “Yes look. You know they've read Sheryl's book. They've leaned in and I'm going to reward them for doing all that."

By the way, that book has very good advice. I'm not knocking Sheryl, I don't want you to misinterpret me. However, you knew there was going to be a however, you have to think about it from the point of view of the employee who did not ask for a raise. They may be doing a better job than the employee who did ask for the raise and in their mind they are going, “Ok, so I didn't ask for a raise and I didn't get a raise. They asked for a raise and they got a raise. What does that mean?"

One, you're not really evaluating people's performance. You're just going, whoever asks, gets. That means I either need to be the guy who asked for the raise, though that's not how I feel. I do my work and I don't necessarily want to ask for a raise. Or I just need to quit and go to a company that actually evaluates performance. You can really make the person who doesn't get the raise feel pretty pissed about it Don't think that when someone is walking through your company doing the "Shmoney Dance," that other people aren't going to notice.

They are going to be fired up about that raise. You can say it is highly confidential that I am giving you this raise. It's not confidential.

The cultural conclusion is that everybody in your company is going to feel that they now have a fiduciary responsibility to their family to ask for a raise all the time because if they don't, they may be missing out on a raise that they would have otherwise gotten. Talk to any experienced CEO and they will tell you this is true. If you give out raises when people just ask you for them, you will have a lot of people asking you for raises. That is called encouraging behavior.

What do you do? The right answer is you have to be formal to save your own culture. I know this is always this is the thing that causes people running startups fits because it's like, " Well I don't want a lot of formalities. I don't want a lot of process. I want it to be organic. We want to do yoga. We want to only smoke organic weed." Sorry that was a Peter Thiel kind of way. Peter got very focused on who was smoking weed a little while ago.

But the process actually protects the culture because what it does it says, look we're going to look at all inputs. We are going to have a formal way of saying anybody who wants a raise, come talk to me. I'm not going to give you a raise but I am happy to hear your story. I'm going to talk to all the people you work with so I get like a understanding. I am going to evaluate all the work that you've done, so I know where I actually rate and what my actual opinion is. I am going to do it periodically, I'm not going to do it daily. If I were fast moving I would do it every six months or even once a quarter. At the end of that process I will tell you what your raise is, I will tell you if you're getting one or if you're not getting one, but I'm not going to do things off cycle. I'm not going to do things when asked. There is one process and that's it.

When I used to be CEO and I had executives, the bigger you become the harder this gets because the more aggressive the people working for you are. To be an executive it turns out you have to often be pretty aggressive. In most companies that's how you get to that level. I would go, “Look, you can lobby me all you want after the process is and I give you your raise, but you know what? I am not hearing it. I already went through my process. I got your input going in. I got everybody else's input. I've got so many people and so much money and you got what I believe is right."

Having a process gets people to be more comfortable because they don't have to always be on edge about, “Am I asking for what I deserve? Or am I getting aced out because of who I am, what I look like? I am not buddy buddy. I'm not at the golf course with you or I'm not doing whatever you like to do? I don't have to worry about any of that because I know your process. You're going to evaluate everybody and then you are going to give them what's fair." That's a much better way to handle that and it means that you're actually understanding what everybody thinks, not just the people you are talking to at the moment.

Now we are going to get into some fun stuff. We are going to evaluate Sam's blog post. There are some very good things in it and there are some things I am going to discuss.

This is the excerpt. "Most employees have only have 90 days after they leave a job to exercise their options. Unfortunately this requires money to cover the strike price and the tax bill..." I'll explain this a little more later, but I want to read it first. “... for the year of exercise. This is often more cash than what the employee has." This is the key. The employee often has to choose between leaving the job and walking away from the vested options i.e. the money that she has because she can't afford to exercise or being locked into staying with the company for all the wrong reasons. So it's a particularly bad situation when an Employee gets terminated. And I'll get into that and that's a really key point. "This doesn't seems fair. The best solution I have I've heard is from Adam D'Angelo, “ a very, very smart guy "at Quora. The idea is to grant options that are exercisable for 10 years from the grant date, which should be nearly all the cases. There's some tricky issues to this." Blah, blah, blah. “But it's still far better than just losing the assets. I think that this is policy that all startups should adopt."

Was Sam right? Is this a policy that all startups should adopt? Let me first explain again what the policy is. Currently, the way almost every stock option package in startup world is this: you get stock that vests over a period of time. When you leave you have, and it depends on the company, 90 days. If you do not buy your stock in that period, it's gone! It's not yours anymore. Which, depending on when you entered the company, could be a big problem. A lot of companies today that have a high valuation, like an Airbnb or an Uber, when they bring you in they go “Wow, if you look at your 409A price compared to the preferred price, the stock we're giving you right now, the options are already worth 10 million dollars." And you're like" Woooo! 10 million dollars. I'm rich."

What they don't necessarily tell you is in order for you to get that money, because the preferred is worth 10 million dollars, your options are probably going to cost you 2.5 million dollars when you leave. If you don't have that 2.5 million dollars in 90 days, it's gone! You just lost all your money. So Sam is like, wow! That's fucked up! And so he wrote a blog post and he said everybody should change it.

The first question that you have to ask yourself is, "This has kind of been around since the 80's, so why has a rule like this been around for 30 years?" It turned out, Sam, I don't know whether he figured this out or just intuited it, but he was right. Something actually had changed. Up until 2004, there used to be a law called APB Opinion Number 25. That law was the old way to account for stock options. It's also the law that all the guys went to jail on. I know a lot of people who caught a case on APB 25, so I'm glad it's gone. It's a very confusing law. A lot people did not understand it and they literally went to jail.

When that was a law, if you gave somebody 10 years to exercise their options, you would never have been able to go public and you would never have been able to be acquired because you were taking an expense that was tied to your stock price. The more your stock went up, the more compensation expense you'd have to take. The worst thing is you wouldn't know what it was going to be. It was totally unpredictable, so you could never forecast earnings. Ever! Because your earnings would be a function of your stock price. The more your stock price went up, the more money you would lose. In those days, people did not look through stock option expenses. It just wasn't doable. That's why everybody's agreement was written at 90 days. That's why it's there. So absolutely it's the right thing to question it being there. Are you guys following? You get this? This is more complicated than the first two examples, but it's a very important one.

Your perspective on this, if you have employees, is you want to be fair. Nobody wants to say to a hire,"Hey you got all this stock in 4 years... SIKE!" Especially when you fire someone. "Hey you're fired! I feel real bad about it BUT guess what? You know, I am also going to take all your money too!"

That's a problem. This is the thing that you have to keep in mind, you have to think about the people who are staying and you want to reward the people who are staying. The perspective of the Employee who leaves, and this is really critical because this is your reputation, is I worked a year, where's my years pay? Now you're telling me about this 90 day exercise? I know it was in the small print of my stock option agreement but my Hiring Manager never told me about that. They never told me I was going to need 2 million dollars to get my stock. Which I don't have. So if I was rich I'd get my stock? That's not fair. So now I am fired and then I'm screwed. Guess what? I am going to tell EVERYBODY how you screwed me over. That's a real reputation problem. That's something that you have to consider for policy.

You also have to consider the employee who stays. One thing that they're going to ask themselves is, look they're leaving and every time anybody leaves it’s like, was that smart? Your Employees know each other better than they know you. In any company, I don't care what company you are. Often the person they're really working with is going to be the person they know more. If that person leaves they're going to go " Well, should I leave too? What did they get and how does that compare to my deal?"

If we look at the situation and analyze it, there are a lot of components. First, companies tread a lot of people around here, the average is somewhere around 10 percent. It's probably getting higher, particularly if you are in San Francisco just because of the culture there. Silicon Valley companies dilute like 6-8 or even 10 percent a year for employee options. You have to keep in mind that as mean as it is, if that employee leaves and can't exercise their options, those options come back to the pool where you can potentially give them to people who are already there. You're actually taking less dilution. That's something that you have to think about. I am not saying you have to act on it but it's something that you have to think about it.

Secondly, losing all your stock is a very big incentive to stay. That could be good news or bad news. It could be good news in that you get to keep somebody you might have lost. It could be bad news in that you kept them for the exact wrong reason: they have handcuffs on. You may get an Employee that is worse than not having an Employee. On the other hand, a 10 year option on a highly volatile security, for those of you who have taken that class, is valuable. 10 years option, volatility and length, that's the value of an option.

10 years on a Startup stock, that's a valuable thing. Remember the employee who stays doesn't get that. The employee who stays just gets a stock. They don't get the new job and the new stock. They get one thing but they don't get both things. You have to weigh that in. This is a hard one. It should be reevaluated by every company. I wouldn't go as far as Sam and say it should be adopted by every company. You have to think about what you want. I would offer two Alternative Cultural Statements. One is, we treat employees with straight forwardness. We're going to be fair and therefore you get 10 years to exercise your stock. What we said we're going to give you, you’re going to get regardless of how rich or poor you are. That's just a done deal.

The second way to handle it - no companies do this, which is why I actually really like this post that he wrote - is you can say up front, " Look you are guaranteed to get your salary but for your stock to be meaningful, these are the things that have to happened. You have to have vested. Two, you have to stay until we get to an exit. Untile the company makes it. You've got other money." Finally, the company actually has to be worth something. Because 10 percent of nothing is nothing. The reason we set the policy this way is we really value people who stay. So don't join this company if you are going to join another one in 18 months because you're going to get screwed. Our policy guarantees you're going to get screwed.

Those are two ways to handle it. It really depends on you and how you want to run your culture. With all these things, it's critical to think it through from everybody's prospective because when push comes to shove, that's going to matter. That's going to change the outcome of your company.

Sam: I am actually revising my recommendation slightly.

Ben Horowitz: Let’s hear it.

Sam: No, it's that I think there needs to be more incentive to stay. If someone gets fired, I still think they get screwed a lot of the time.

Ben Horowitz: The other thing that's really important, that Sam pointed out, is how much money you have. If you have the money, you don't get screwed. You can buy your stock. You do take some risks, but you can buy your stock. If you don't have the money, you don't have the money.

Now we're getting to the person on my shirt: Toussaint. He was the best at this and I want to take you through some examples because they're very powerful. Toussaint was born a slave. He wasn't just born a slave, he was born a slave in the most brutal place to be a slave, which was the Colony of Santa Domingo, now known as Haiti. This was a much more severe form of slavery, as were all the sugar growing areas, then in the US, which was historically a very brutal form of slavery. To give you some numbers, over the course of slavery, somewhere like 400 years, a million slaves were brought to the US. At the end of slavery, there were four million slaves in the US. In that same period, in the sugar growing countries in the Caribbean, two million slaves were brought over and at the end of slavery there were seven hundred thousand left. From just a quantitative perspective, nearly 10 times more brutal. I am going to read this to you. I don't know if I quite have time but I don't care. I'll read you a description of slavery in Toussaint's area.

Whipping was interrupted in order to pass a piece of hot wood on the buttocks of the victim. Salt, pepper, citrus, cinders, aloes and hot ashes were poured into bleeding wounds. It's not to heal them. This is to make it worse. Mutilations were common. Limbs, ears and sometimes private parts to deprive them of the pleasures which they could indulge without expense. Their masters poured burning wax on their arms and hands and shoulders. Emptied the boiling king sugar over their heads. Burning them alive. Roasting them on the slow fires. Filled them with gun powder and blew them up with a match. Buried them up to their neck and smeared their head with sugar that that the flies might devour them. Fastened them to the nest of ants or wasps. And made them eat the excrement, drink the urine, lick the saliva of other slaves. One

Colonist was known `in moments of anger to throw himself on a slaves and stick his teeth into their flesh.' That's the slavery that he grew up in. It's really important to understand this because to get out of that perspective was not easy. But he had a vision that was threefold. One, he wanted to end slavery. Two, he wanted to actually take control of the country and run the country. And three, he wanted it to be a world-class country. Not one in which he had simply freed the slaves but one that could compete on a worldwide basis. That was his mindset going in. What I read was the environment he came from.

A management example is conquering the enemy. The sequence of battles that occurred in Haiti were, he first had to defeat the locals. Once he defeated the locals, there were several countries that were very, very interested in taking control of Haiti. Principally Spain, England and France. He had to defeat those armies as well. When he conquered them, he had to decide what to do with the conquered soldiers and the leaders on the other side. He took into perspective three different points of view. One, his soldier’s point of view. Two, his enemies point of view. And finally ,the point of view of the resulting culture. What kind of country was he building? The army was going to be the seed corn for the culture of the whole country.

From the soldier’s perspective: do we get to pillage? Soldiers like to pillage. It’s something for their work. The second thing is they're trying to kill us so we should kill them. That's a basic perspective of the people who are fighting for him, the most important people to Toussaint. I put pillage up there, a couple of things to know, one I didn't put rape up there. Very interestingly, not only did he not allow rape among his army, but he didn't even allow his officers to cheat on their wives. If they did he would get rid of them because he was so concerned about the resulting culture. What was it going to be? Was it going to be productive? Was it going to be the best in world? Or was it going to be something less than that? That was his mindset going in. His army was actually famous for not pillaging. They were already actually used to this. This was one of the most surprising things to the conquered people to the point where where even the white people were very impressed because he would go into their city and not pillage even though he would win. Again, this is because he took a long view of the culture.

This is an important subtle point. He believed the culture of Haiti, because it was a slave culture, sugar plantation culture, was pretty low grade compared to what he had experienced in Europe when he dealt with the Europeans. He thought that slave culture was even more broken than Haitian culture because it's the kind of culture where, "Oh you don't do what I tell you? I'm going to beat you to death. I am going to blow you up with gun powder." If you think about the behavior that ensues from that, that was the culture he knew he needed to replace. He knew he needed to upgrade.

His solution when he conquered the British, or the Spanish, or he conquered the French, was he would take the very best people from the opposing side and he would make them generals in his army. You probably didn't expect that. Here are the guys trying to kill him. He's leading a slave evolution and when he conquers the enemy, he actually incorporates them into his army and makes them part of it. He wanted the expertise and he wanted the culture to be at a much higher level.

The second question he had, this even more complicated, was what do you do with the slave owners? You're leading slave revolution, you take control of the country,what do you do with the slave owners? Three perspectives again. For the slaves, you want to kill the slave owners. There is no question. That's your land now. You won. F them. From Toussaint's perspective, it was more complicated because he wanted Haiti to be a first world country and sugar was really important. The whole slave economy was the sugar economy. On the other hand, he was a slave and he had to have been pretty upset. Particularly given the type of slavery. But he didn't know how to run a sugar plantation and he didn't have any business relationships for trading sugar. So what to do? If you look at the slave owner perspective, it's pretty interesting because they're coming at it - and this is the point of view that he actually had the discipline to understand - they were coming from a cost structure that was predicated on slave labor. Their business didn't work without slave labor. If they had to pay people their cash flow wouldn't work. They paid a lot of money for the slaves up front and they paid a lot of money for the land. In their mind, that was how business worked. You can't just change the economics and have it still work. They knew they had some power because of the position they were in.

So what was the answer for the slave owners? The solution was one, to end slavery. Two, let the slave owner keep their land. Three, make them pay their workers. There was no more slave labor. In order to fund that, lower their taxes. You guys ought to be kind of impressed with that.

Lower the taxes of the slave owners after you defeat the slave owners and end slavery. But the bigger goal was he wanted a stronger culture. The way he treated those slave owners, the need to keep the economy going was important. Let's look at the results. First of all, Toussaint's revolution is the only successful slave revolution in the history of mankind. There has never been another one and, hopefully we won't have slavery in a big way, there won't be another one. He's it. Two, the plantation owners kept their land. Three he defeated Napoleon. He had a booming economy and a world class culture. Under Toussaint, Haiti had more export revenue than the United States. That's how successful he was in the revolution. This is the power of looking at a situation not just from your point of view, but from the point of view of all the constituents. Even the people you hate. Which is hard to do when you are a CEO and harder to do when you are leading the revolution.

In conclusion, the most important thing that you can learn, and one of the hardest things to do, is you have to discipline yourself to see your company through the eyes of the employees, through the eyes of your partners, through the eyes of the people you are not talking to and who are not in the room.

Thank you.

Now I will take questions.

Q: If you have to fire or demote an executive, how do you have the conversation and then how do you explain it to everyone else?

Ben Horowitz: Right, this is great question. Clearly it's some kind of failure. You failed on hiring. You failed on integrating. They failed at their job. The first thing when firing the person is to really try to be honest. You're feeling like you failed. Common reactions are, you suck and so I am firing you. Screw off. That's not good because it's not really true. You may be feeling that way. Another common mistake is to be too mushy. It's not you it's me. This feels like a weird break up with an ex-boyfriend that you really didn't like.

Generally when you hire people, you try to hire the very best. You hire people who are qualified to do the job. The reason they fail on the job is you made some mistake in the hiring process and you didn't match them to the needs of your company accurately enough. That's the number one reason why this fails and so that's generally a good place to start. To say look, here's how we are and here's what I didn't recognize about us and about you when I made the decision. It is what it is. We're going to have to move on.

When you talk to the Employees about it, it's different. You can take somebody's job, you have to take their job, but you don't to take their dignity. This is something Bill Campbell taught me. It's not necessary to get up in front of the company and say, "I blew that mother fucker out. I capped his ass."

In fact it's not good. Nobody feels good about that.You might feel proud of yourself but nobody else feels good about that. The right thing to do is thank them for their work. Like let people know that they're moving on. You don't have to explain all their personal details. It's more important to leave them with their dignity and let them go on to live another day. What you say at that meeting is their reputation, because everybody in your company is going to call on that person when they try and get their next job. If you start saying a bunch of BS about them, that's not going to be good and it's not going get interpreted as we screwed up, it's going get interpreted as he screwed up. You have to be very honest with them but you have to make sure you preserve their dignity when you talk to the company.

Q: I was reading your book yesterday. How did you deal with all the stress? Was it meditating? Hip Hop?

Ben Horowitz: The answer is I used to be 6 foot 4 and good looking so clearly not very well. I get asked that a lot and I have a great answer for it. I have a wonderful wife who is sitting right here. I will oar and he borrowed that technique from him but applied it in a much kind of more dramatic context. He had British, French, Spanish, slaves and mulattoes, most of the mulattoes in Haiti at that time were pro-slavery. That was another issue, but his leadership was so great, everybody wanted to join him.

Q: How do you incorporate Toussaint's ideology and get people who were previously against you on your side?

Ben Horowitz:What he did in general is the right thing. You have to show them a better way, as a leader, if somebody's your enemy and you need to convert them. This happens in business too. Somebody is a competitor and you want to bring them over but you don't want to bring in, ethically, people who switch from one competitor to another. Your culture has to be elevated, your mission has to be elevated. Your way of doing things has to be just better. That was what was so compelling for the rest of the army.

Q: I am curious to learn how you have built a culture around people and among the entrepreneurs that you work with that has differentiated you in the market from all the other venture capital firms.

Ben Horowitz: Probably not the best question for me. I can ask Sam that. I don't know. The question is how we do build a culture out at Andreessen Horowitz that is differentiated us from all other VC's. I feel like that is certainly the goal. We have been around for 5 years now. The attempt that we made, it's for the rest of the world to judge if we succeeded, was this: in the old days of VCs, when I was a entrepreneur, the basic idea was you have an entrepreneur or inventor and they get a company to a point and at that point they either are ready to be CEO or you would go find a CEO to replace them and build "the company." Our cultural philosophy is that the founder and inventor are special. We're going to design the firm and the culture of the firm to help the founder develop into a CEO. We do a lot of systematic things different. The two biggest are all of our partners are founders or CEOs. It's an original model where some sort of experience is required. That's a joke.

If you are an advisor to a CEO you have to have have been a CEO. Imagine that. That's why I like Sam, he used to be a CEO. He doesn't talk about it that much but he was a CEO and was good at it. The second part is that a professional CEO will bring in, in the old days, a network of people that knows. Guys who brought technology in big corporations to important partners in the field to people in the press. We try to build that network on your behalf at the firm. I think we do a better job of that than anybody else. Those are the ways that we try to be different.

Q: Putting yourself in other people's shoes is very important. Can you give us some tips?

Ben Horowitz: Putting yourself in other people’s shoes is difficult in management. It's hard in daily life. It's even harder in management because it's the stress at the moment. If a great employee is asking you for a raise, it is very hard for not to respond because you do not want to lose them and they are not asking you for a raise randomly, they are asking you for a reason. If you don't have a process in place to go stop back out, you have to pause yourself. If somebody comes to you with something that you know is important, you want to feel like you have all the answers. Right now you guys are asking me questions and if I don't know the answer I will make something up because I want you guys to think I am smart.

The most importan thing is to pause. If you know something is really important and you haven't thought it through, just to say, "I am taking this really seriously but I have to pause because I have to think it through from all perspectives. I’m going to come back." I end up doing that a lot just because there are a lot of things that you run into that you have never seen before. Most CEO's, including myself, learn this the hard way. You go ok, I'm going sneak away with this. Nobody's going to see me give them the raise. I'm going to do it and it’s going to be all under the covers. Confidentiality baby. Then it blows up in your face three weeks later and you're like, "Oh my God. What have I done?" Or three months later or even a year later. Then once it's a year later, it's a huge problem.

You've taken what was a little emotional problem and you've turned it into a forest fire. We call it a Kimchi problem. The deeper you bury it the hotter it gets. It's a Korean joke. It takes practice. It’s very difficult to do. My friend Bill Campbell, this is his big skill. People always trying to describe him to me and I’m like that is not him at all. That’s not what he’s good at. He’s good at seeing the company through the eyes of the employees. If you are good at that, you will very likely be an elite leader.

Thank you.

Lecture 16: How to Run a User Interview

Emmett Shear

Sam Altman: Good afternoon. Today's guest speaker is Emmett Shear. Emmett is the CEO of Twitch, which was acquired by Amazon, where he now works. Emmett is going to talk about how to do great user interviews; this is the talking to users part of "How to Start a Start Up." It should be really useful. Thank you very much for coming!

Emmett Shear: Thanks Sam. I started my first startup with Justin Kan right out of college. We started a company called Kiko Calendar. It didn't go so well. Well, it went alright. We built it, we sold it on eBay. That's not necessarily the end you want for your start up.

It was a good time. We learned a lot. We learned a lot about programming. We didn't know anything about calendars. Neither one of us were users of calendars. Nor did we, during the period of time we did the thing, go talk to anyone who actually did use a calendar. That was not optimal. We got the build stuff part of the startup down. We did not get to the talk to users part.

The second startup we started, we used a very common trick that lets you get away with not talking to users, which was that we were our own consumer. We had this idea for a television show, Justin.tv, a reality show of Justin Kan's life. We built technology and a website around the reality show we wanted to run. We were the users for that product. One way to cheat and get away with not talking to many other users is to build something that literally is just for you. Then you don't need to talk to anyone else because you know what you want and what you need. But that is limiting way to start a startup. Most startups are not built for the person who is using them. When you do that, every now and then you get really lucky and you are a representative of some huge class of people who all want the exact same thing you do. But very often, it just turns into a side project that doesn't go anywhere.

We kept working on Justin.tv for awhile and we actually achieved a good deal of success because it turned out that there were people out there who wanted to do the same thing we did, which was broadcast our lives on the internet. The issue with Justin.tv that kept us from achieving greatness is we hadn't figured out how to build towards anything beyond that initial TV show. We built a great product. If you wanted to run a live 24/7 Reality television show about your life, we had the website for you. We had exactly what you needed but if we wanted to go do more than that. We wanted to open it up to a broader spectrum of people and use cases, but we didn't have the insight to figure that out because we weren't that user.

So we decided to pivot Justin.tv. We decided we needed to go in a new direction. We thought we built a lot of valuable technology but we hadn't identified a use case that would let it get really big. There were two directions that seemed promising. One was mobile and one was gaming. I lead the gaming initiative inside of the company. What we did with gaming that was very, very different from what we'd ever done before was we actually went and talked to users. Because while I loved watching gaming videos, I was very aware that neither I nor anyone else in the company knew anything about broadcasting video games. I was amped about the content. I thought that there was market there. That was the insight that wasn't common at the time, which was how much fun it was to watch video games.

Quick show of hands, how many people know about watching video games on the internet here? If you don't know about watching video games internet you should go read about it, because it's important context for the stuff I am going to talk about. The main point is I thought it was awesome, but I didn't know anything about the important side of it, which is actually acquiring the content of the startup broadcasting. We ran a very large number of user interviews. We talked to a lot of people and that data formed the core of all the decision making for the next three years of product features on Twitch. We continued to talk to users and in fact built an entire part of the company whose job it is to talk to our users. That is a whole division we didn't even have at Justin.tv. We had no one at the company whose job it was to talk to our most important users.

I want to give you guys a little bit of insight into what it meant to talk to users. We determined that the broadcasters were the most important people because when we went and looked into the market, we looked into what determined why people watched a certain stream or went to a certain website. They would just follow the content. You had a piece of content you loved and the broadcaster would come with you. That's actually the one really important point about user interviews, which is that who you talk to is as important as what questions you ask and what you pull away from it. Because if you go and talk to a set of users, if we had gone and talked to viewers only, we would have gotten completely different feedback than if we were talking to the broadcasters. Talking to the broadcasters gave us insight into how to build something for them. That turned out to be strategically correct. I wish I could tell you the recipe for figuring who the target user is for your product, and who your target user should be, but there isn't a recipe. It comes down to thinking really hard and using your judgment to figure out who you are really building this for.

I want to do something interactive now. I'm going to get a bunch of ideas from you guys and I'm going to pick one of them. I want everyone to sit down and do step one of this process right now. Which people, where would you go to find the people you needed to talk to in order to learn about what you should build. The idea we are going to use is a lecture focused note taking app. The idea is: the state of the art for note taking is not good enough yet and I want to make a note taking app that improves that experience. It will make taking notes in class better. Maybe it has collaboration features or maybe it helps you focus better somehow. It has multimedia enhancements. All sorts of possible features. That's the idea. So take 120 seconds right now and think about not what you would ask or what the right features for this app is, but who would you talk to? Who is going to give you that feedback that is going to tell you whether this is good or not. It’s good to think in your head but actually write it down and come up with the five people you would talk to. The five types of people you talk to, and who you think the most important one was. There's nothing like actually running through something and trying to do it. Actually get it into your head that it's right way to do it. I'm gratified to hear the clicking of keyboards now. If you are following along at home actually do it. Think about who would you talk to because that's the first question for almost any startup. You need the answer to the question: who is my user and where am I going to find them?

Alright, that's shorter than you normally would think about this problem. It's actually a really tricky problem, figuring out where to source people is pretty hard. We’re going to move along in this highly abbreviated version of learning how to build a product and running a user interview.

Can I get one volunteer from the audience to come up and tell us who you would talk to. And we'll talk about it.

Audience Member 1: I would definitely talk to college students first, because we sit in a lot of lectures. Specifically, I want to talk to college students studying different subjects to see if they are an English major, if that makes a difference versus studying Math or Computer Science in terms of how you want to take notes in different lectures.

Emmett Shear: You're going to talk to a bunch of college students. Would you pick any particular subset of college students? We don't want to talk to all college students.

Audience Member 1: I want to only talk to college students and break down the divisions by people who study different areas. It would make sense for people who have different study techniques, because some people take a lot of notes. Some people don't take that many notes but still jot stuff down.

Emmett Shear: That's a really good start. Those are obviously the users you want to go talk to, especially if you are targeting college students as the consumer. If you are talking to college students as the consumer, you are going to get a lot out of students about what their current note taking habits are and what they would be excited about.

One of the problems with selling things to college students is that college students actually don't spend very much money. It's really hard to get you guys to open your wallets, especially if you want them to pay for a school related thing. People don't even want to buy text books. You probably all use checks or debit or borrow from your friends. So one of the things that I think you would be missing if you go after just the students, is who the most important person to this this app is. If you actually had a note taking app for colleges, the people most likely to actually buy a note taking app would be college IT.

Presumably for the most part you want to sell software to students, and the people who have to be brought into that is usually the school administrators. That might be one approach. I feel like you will presumably go talk to college students and find out they don't actually buy any note software now at all. It's possible they do, in which case I'm completely wrong. This is why you have to go and talk to the users.

But you then have to try other groups. So I would talk to college IT administrators as well.That's another area that's really promising. You might talk to parents. Who spends money on their kids' education? Who is willing to pull their wallet out? The parents of kids who are freshmen going to college for the first time. You need this app to make your kid productive so that they don't fail out of college. There are actually a lot of groups that aren't necessarily obvious users but who are potentially critical to your app's success. When you are at the very beginning of a startup, when you have this idea that you think is awesome, you want to have the broadest group you possibly can. You don't just want to talk to one type of person. You want to get familiar with the various kind of people who could be contributing.

Let's have someone come up and we're going to pretend we are running a user interview. We are going to talk to a college student and try to find out what we should build, what we should get into this note taking app. Another volunteer please, for running an interview. Hello.

Audience Member 2: Hi, I'm Stephanie.

Emmett Shear: Hi Stephanie.

Audience Member 2: Nice to meet you.

Emmett Shear: Welcome. Thank you for agreeing to do this user interview with us. I want to hear from you about your note taking habits. How do you take notes today?

Audience Member 2: Sure, I take notes in a variety of ways. Because of speed and efficiency, and because I can come back to it later, it's easy for me to take notes on my laptop. A lot of those would be primarily text based, but in certain classes, for example if I am taking a History class, most of it would be in text. But if I am taking a Physics class, there are going to be more complex diagrams and different angles I have to draw.

Emmett Shear: What software do you use for this stuff today?

Audience Member 2: I just do pen and paper for that.

Emmett Shear: You do pen and paper. So you do a combination. You take notes with pen and paper. You take notes with the computer sometimes.

Audience Member 2: Yup.

Emmett Shear: When you take all these notes, at the end, do you actually review them? Be honest! Do you actually go back and look at this notes?

Audience Member 2: The pen and paper not so much. But yes to the software based. It's easier to access and it's easier for me to share and collaborate and maybe even merge notes with classmates and friends.

Emmett Shear: What did you use to take notes today on your computer?

Audience Member 2: Google Docs and Evernote.

Emmett Shear: Why two things at the same time?

Audience Member 2: Evernote is easy if I am trying to just collect for myself. You can share, but Google Docs is easier to share. If a friend has already created a folder in Google Docs, I just have to add to that folder. If it's for my personal use I tend to go more toward Evernote.

Emmett Shear: It sounds like you have a lot of note taking collaborations.

Audience Member 2: Yeah, I wish it was integrated.

Emmett Shear: Tell me more about that. Do you wind up taking most of the notes, most of the value out of notes that other people take? Or is it mostly your own notes you review at the end of the semester? How does that work?

Audience Member 2: It's mostly mine because I am pretty picky about the way I like things organized. Design wise or formatting, even color, I am really particular with. The font that we use really effects the way I study. So I tend to personalize notes, even after I merge them.

Emmett Shear: So you're pulling notes from other people but then you merge them into what works for you. Awesome! If you have Evernote notes and you have Google Docs notes, and you have pen and paper notes, once the semester is over, do you ever go back to any of that stuff or is it per quarter? Once the quarter is over, do you ever go back to any of that stuff?

Audience Member 2: For classes not so much, but if it's notes that I have taken for talks, like these for example, or if it's interview prep, I tend to go back because I like to keep these things fresh in my mind. They help me prep for future things.

Emmett Shear: That's interesting. Tell me more about that. You take notes not just in class.

Audience Member 2: I take notes to summarize main points. For example, inspirational quotes from talks like these. If I am going to an event where I am going to meet someone, notes help me remember what was at the talk.

Emmett Shear: Awesome. Normally I would actually dig into a lot more detail. There are a huge amount of open questions that are still in my mind after hearing that. Which people do you collaborate with? How long are your notes?. How much time do you spend note taking? I would dig into her current behavior but in the interest of time and not making everyone hear about the intricacies of one person’s note taking habits forever, we're going to move on. Thank you very much Stephanie.

Audience Member 2: Thank you.

Emmett Shear: I appreciate that. You notice we are not talking about the actual content of the app at all. I'm not really interested in features. I don't want to know about a specific feature set in Google Docs or Evernote. I might dig a little more into which features actually get used. If she's actively collaborating, how does that work? I heard some interesting things, " We use a folder." That's interesting to me.

The main thing you're trying to do when running this first set of interviews is not necessarily ask questions about optimizing user flow. Or questions about the specifics of any of that stuff. That can be distracting because users think they know what they want. You get the horseless carriage effect where you're asked for a faster horse instead of asked to design the actual solution to the problem.

So you want to stay as far away from features as possible because the things they tell you feel overwhelmingly real. When you have a real user asking you for a feature, it's very hard to say no to them because here's a real person who really has this problem. They're saying, "Build me this feature." But as you start to talk to lots of people and really get a sense for what their problems are, you figure out if this is actually a promising area or not.

Based on what I heard there, starting from that user interview, I'm not necessarily positive there is a problem. At least there might not be a big enough problem that it's worth building a whole new product for. I didn't hear a lot of things that were big blockers, where there is something really wrong with the way it was working. Unless I have some big idea, I would take that as a negative sign. That doesn't necessarily mean you can't move forward and keep talking to more people. Just because you don't get anything out of talking to the first person doesn't mean there are not going to be more people who actually have a problem. Once you've talked to about six to eight people, you are usually about done. It's unlikely you're going to discover a bunch of new information. Which is why it is important to talk to different extremes of people. Go find people who are different, because if you talk to six Stanford College students you are going to get a very different response than if you talk to six high school students or six parents.

Based on that though, you can come up with a set of ideas.You have this information about how someone takes notes. You had some ideas as to how you could build something cool. If you are going to build just one feature on top of Google Docs, what would that feature be? For a new product like this, it might be a good way to start thinking about where to go. They are extensively using this thing right now, how can we make that experience just one quantum better? Something that would be really exciting to this person, something that would be one step ahead. Take two minutes and think about what that feature might be. Try to come up with what you might do based on what you heard from Stephanie that could convince her to switch away from her current collaborative, multi-person, all working together work flow on Google Docs to your new thing that has the features of Google Docs plus this one special thing that is going to make it more useful and convince them to stop using the thing they are already using.

Awesome, alright. I am going to invite our third guest up.

Audience Member 3: The reason she uses Evernote is because of sticky note type notes. More thoughts and like details. I feel that Google Docs has documents and not smaller notes. I feel like a feature that would be super would be a mobile version of drive that isn't clunky and doesn't make you use documents could be useful.

Emmett Shear: Awesome. That's a good insight. That's exactly one of the things you get out of that user interview. Now you have this idea. You've gotten this user’s feedback. What if we had a Google Docs that had the collaborative aspects and the group aspects but where you could pull in more little one off notes. A product designed more around note taking. The question is now, once you have this idea, is this enough? Is this something people would actually switch to? There are two ways to validate that. One, if you are quick at programming you can literally just go build it, throw it out into the world, and see what happens. When that works, it's an excellent way to approach the problem. But a lot of the time that one little thing that's a bit better might take you three months to actually build. So you want to go out and validate that idea further before you start building it.

You might take that idea and draw diagrams of what it would look like. Draw the work flow and put that in front of people. The one thing you really don't want to do is ask them about a great idea for a feature. Ask them, "Are you excited about it?" Because the feedback you get from users if you tell them about a feature and ask them, "Is this feature good?" is often, "Oh yeah that's great." When you actually build it, you find out that while they thought it was a clever idea, no one actually cares to switch and get it. So the one question you can't ask is, "Is this feature actually good or not?"

Sam Altman: What is the minimum that you can do to actually build on that question? Between asking and actually building the full thing?

Emmett Shear: What's the minimum you can actually get away with to validate your product, given you can't actually just go and ask them, "Is this good or not?" It's highly dependent on the particular feature. Usually the best thing you can do is just hack something together. If your idea is to build something on top of Google Docs, don't go rebuild Google Documents but for note taking. Find a way to write a browser extension that stuffs that little bit of incremental feature in and see if it's actually useful for people. Find a way to cheat is what it comes down to, because if you can't actually put it in front of people it's really, really hard to find that out.

For bigger things, where you are actually trying to get people to spend money, it gets a lot easier. If you are selling, it's great. Sales is the cure-all for this problem. Get people to give you their credit card and I guarantee you they are actually interested in the feature. It's one of the most validating things that you can do for a product. Go out there and actually get customers to commit to pay you up front. The problem when you are working on a student note taking app, is that's going to be relatively hard. Unless your idea is that you're actually going to sell it, the trial version is probably free. You're not necessarily going to learn that much by trying to charge people money. But if you go out there and can get people to say "Hey, I am going to give you money," the money test is amazing. It clarifies whether or not they're really excited about your product. If you're not five dollars excited about it, you're probably not very excited about it.

The last thing I want to do is work through what happened at Twitch. I brought some slides that I'd like to put up. They are representative excerpts of Twitch feedback. I had a twenty-six page document of all the feedback and realized that reading that was going to be a little bit tedious. Lots of people said this to us when we asked them questions. I've pre-condensed the feedback for you.

To launch Twitch, we talked to a bunch of existing Justin.tv broadcasters and asked them about their experience broadcasting, what they liked about broadcasting, why they broadcasted, what they broadcasted. What else was going on in their life? When you talk to detailed users of your product, they come back to you with very detailed things about features because they get mired in the features. You have to sort of read between the lines. They ask us for things like, “I want a way to clear the ban list in my chatroom." That was actually a very common request because there was a very particular issue with how our chatroom is worked. People would ask for the ability to edit titles of highlights after creating them. This stuff was really consistent.

As we talked to broadcasters, we probably talked to twelve to fourteen broadcasters of the Justin.tv gaming platform, we got all this feedback. "Your competitors have all these cool features like polls and scrolling text. I can personalize chat there." Then we have some positive feedback. "You guys don't have ads. You're able to ban trolls." A bunch of stuff about chat, around interactivity with their viewers. That was all really interesting. This is what the Justin.tv broadcasters wanted us to build. This is where they felt pain using the product. If you thought that what we did was go and address these problems, you would be wrong. People who are using your service already are willing to put up with all these issues, which kind of means that these are probably not the biggest problems. If you are willing to ignore the fact that you can edit the ban list and that titles are editable, that there is no way to get trolls out of your channel and you're using the service anyways, maybe those aren't huge problems. That brings up a really important point, which is you have to compare groups of people. And compare the level in which they argue.

Here we have competitor broadcaster feedback, which is really interesting. This is stuff we heard a lot from people who were using other broadcast platforms. They wanted to be able to switch multiple people onto their channel at the same time. They complained about us not having a revshare program. They talked a lot about how they were trying to make a living and they really wanted to make money pursuing this this gaming broadcasting thing. They talked about video stability. Our service wasn't good in Europe. Globally, video stability was this huge, huge issue. If you compare and contrast, it was really different. What people who didn't use our service cared about was completely different than the people who were using our service. We focused on this stuff because this was the stuff that was so bad that people weren't even willing to use our service. Most of it we hadn't actuallythought about it because our user base happened to be well educated and knew about all their options. Reaching out to them meant that they probably already tried all four services and actually had an opinion. It's great when you can get users who are that informed and understand the space that well.

The other important thing we did was talk to non-broadcasters. We talked to all the people who weren't using us or our competitors. In many ways, those were the most important people. Talking to your competitors is a a short term win, unless your software is like Google, which is a search engine which everyone uses, then there may be no non-users to convert. In the case of gaming broadcasting, almost everyone is a non-user. The majority of people you are competing with are non-users. They are people who have never used your service before and what they say is actually the most important. What they say is the thing that blocks you from expanding the size of the market with your features.

If all you do is look at your competitors and talk to people who use your competitors' products, you can never expand. You're not learning things that help you expand the size of the market. You want to talk to people who aren't even trying to use these things yet. Who've thought about it maybe, but who aren’t into it. What did they say? My computer isn't fast enough. I am focused on training twelve hours day for the next tournament. I like making the perfect video and I like editing it. I upload a couple of things to YouTube. I don't do live streaming. I have no desire to go into that space. In Korea this is a big problem. Once our strategy gets broadcast in major tournaments, we have to start over. We have to come up with an entirely new strategy. The last thing we ever want to do is broadcast our practice sessions, are you crazy? That's going to hurt us in the next big tournament.

This became big outreach program for us, trying to figure out how to get people over this. We brought people computers. We worked closely with gaming broadcast software companies to help the people who made the broadcasting software. We started building broadcasting into games and into platforms. We built broadcasting into the Xbox. We brought broadcasting into PlayStation 4 because we needed to overcome this issue. Broadcasting wasn't possible. These were the three big groups we looked at for broadcasting. You combine that feedback and what it tells you is not features to build, because the features they asked for, polls, the ability to have a child account, we haven't built most of that stuff. What was important were the issues, the goals they were trying to accomplish.

People wanted money. People wanted stability and quality. People wanted universal access for viewers all around the world. That became our focus. We dumped almost all of our resources into things no one ever mentioned in an interview. Those were the things that actually addressed the problem. The way that you can tell that it worked is as we would build these things, we would go back to exact same people we interviewed and we would say, "You told us you really cared a lot about making money. We built you this subscription program that will let you make money."

It's astonishing because most people had actually never had that experience. They had never talked to someone and said, "It would be really great if your product had feature X" and then a month later your product actually has feature X, or at the very least a feature that addresses the problem that they brought up. The people we converted first to our product were the people that we talked to about user research. They were the ones who were the most impressed. Which is fun. It really worked, because we picked people who were representative. We picked big broadcasters. Small ones. Medium ones. We made sure were addressing their concerns. That was completely different from how we approached the problem at Justin.tv.

With Justin.tv when we tried to do this, we'd go through huge amounts of data. We spent tons of time looking through Google Analytics, Mixpanel, and in-house analytics tools. Figuring out how people were trying to use the service, where our traffic came from, completion rates on flows. You can learn things from that. I'm not telling you not to look at your data. But it doesn't tell you what the problems are you need to address. We would invent these ideas at Justin.tv without talking to someone and then nine times out of ten, that idea would turn out to be bad. That’s actually one of the disappointing things about doing user interviews and getting user feedback, which is why I think so many people don't do it. You're going to get negative news about your favorite feature most of the time. You're going to have this great idea and you're going to talk to users and it’s going to turn out that nobody actually wants it. They are actually concerned about a completely different set of things and they don't care about what you thought was important at all. That’s a little bit sad, but think about how sad you'd be in four months when you launch that feature and it turns out no one actually wants to use it.

That's the lecture section. I want to take some questions from the audience.

Audience Member 4 : What do you see startups get most wrong about interviews? Most startups don't do them at all, but the ones that do, what are their most common mistakes?

Emmett Shear: The most common mistake is showing people your product. Don't show them your product. It’s like telling them about a feature. You want to learn what's already in their heads. You want avoid putting things there. The other thing is asking about your pet feature direction. If you think you want to add subscriptions to your product, going and asking people, "Would you pay for a subscription? Would you use this feature?"

Another big mistake people make is talking to who is available rather than talking to who they need to talk to. There are certain users are really easy to get at because they are members of your forum already. You have some product forum, you talk to the users on that forum because they’re easy to access. We spent weeks digging for identifying information and figuring out who these people were so we could talk to them. This was a site that did not support messaging, so there was no obvious way to interact with them. We spent a bunch of time trying to network and find those users. Because if you just talk to who's easy to talk to, you're not getting the best data. The fortunate side there is almost everyone is flattered to be asked what they think, so they will actually talk to you and tell you things.

Audience Member 5: How hard was it to get buy in from the rest of your company? You can say, "I'm in charge so you're doing what I say" but that's probably not the best way of doing it.

Emmett Shear: That's a good question. If you just go to them and say, "I talked to the users. I figured it out. We have to build this," it's really hard because people don't trust you. There's something magic about showing them the interview though. I recommend recording interviews. It also stops you from taking notes in the middle, which is a little bit disruptive. It makes it hard for you to actually engage in the conversation. You can then play that recording for people. They don't have to be there for the entirety of all the interviews, but when you want to make a point about what what you should be building and why, you can play the interview back for the rest of the company. It's like magic, the influence it has on people's thoughts, on what is the right thing to build.

Audience Member 6: Since you mentioned recording, did you try to insist on doing Skype interviews rather than over email?

Emmett Shear: You definitely want to Skype. You don't want to do interviews over email if you can avoid it, because interviews over email are non-interactive. The most interesting learnings come from the, “Interesting. Tell me more." The instant you hit this vein, they will say something that you didn't expect. And then you should drop into detective mode. Detective mode is, "Huh, that's interesting. Can you tell me more about that?" People don't like silence, so they'll keep talking to feel the void. The best part about doing an interview over Skype or doing it in person is that you have that interactive feedback. You can actually pull a lot more out of people. Email interviews are basically useless. In person or over Skype interviews are also easy to record. Make sure you ask them if it's ok to record. It's not polite to record people without their consent, but if they are willing to give you a user interview, they'll probably willing for you to record it as well.

Audience Member 6: What about the international market? You mentioned that you have a lot of users in Korea.

Emmett Shear: That's really hard. To this day Twitch works way better in English speaking countries than it does in non-English speaking countries. A big part of that is we are much better at talking to people in English speaking countries and learning what their needs are. We are not as good at that in other Countries. We've tried to address that by hiring people who speak Korean and having them translate. We've tried to address it by finding representative people who speak both English and Korean and reaching out to them. But the problem is you're not actually getting a representative sample, no matter how hard you try. The very fact that they are a fluent English speaker means they are not representative of all the people who don't speak fluent English. It's a hard problem. It’s why companies find it easier to build markets in their home country. It's really hard to talk to users abroad.

Audience Member 7: What channels do you use to reach out to them? And do you ever compensate them?

Emmett Shear: The channels we used to reach out to them were onsite messaging systems. Most websites have some way to contact the user. If they are a visible user of another website, you use that site's messaging system and say, "Hey. I was watching your stream...” Or, "I'd love to ask you some questions about your usage. Would you mind hopping on a Skype call?" We also find out where those people were. We'd run into them at events because a lot of these people go to the same events. We wouldn't run the user interview at the event, but we'd get to know them. We would exchange business cards and we would get in touch with them. We tend not to compensate people. If people don't care enough about the problem to like someone who is trying to solve it, you're probably barking up the wrong tree. We never had any trouble getting people to talk to us without paying them.

Audience Member 7: What about onsite user feedback tools?

Emmett Shear: This is a whole second set of user feedback that's really important. You're talking about when you have a new product and you want to see if it's actually going to work or not. You put it in front of people and see how they use it or not. That is super, super important. It can tell you where you went wrong building something before you launch it, which is great. It doesn't tell you what to build. It helps you iron out the kinks and edges of the thing you did build. Generally speaking, that wasn't the user feedback we were getting. I mean that stuff's good, it's much more similar though to the data driving approach. You're finding out why people are dropping off of this flow. You're not finding out the problem you should really be solving. What do they care about as a human? This early stage user interview is crucial for startups. That's where you want to focus. We didn't bring anyone in onsite, it was almost all over phone or Skype.

Audience Member 8: In finding groups of people that can give different kinds of feedback, is there a group that you should focus on first?

Emmett Shear: Given that we had very limited resources, we focused on the people using competing products. We knew that they were already interested in the behavior that we needed and they were willing to do it at all. Therefore all we had to do was convince them to switch, which is much easier to do than to create a new behavior. We did that because we had to get some quick wins. My gaming project inside of Justin.tv would have been killed if wasn't showing twenty-five percent month over month growth every single month. That meant focusing on the short-term, on getting the people in right now. That turned out to be good in general.

Audience Member 9: In the beginning the video gaming industry was decentralized. There wasn't a lot of cohesion, but now it's very different. You said you originally spoke to broadcasters and streamers themselves. How has that changed? For example, ? has banned users or professional players from streaming their own stuff. Did you try to gain leverage with that?

Emmett Shear: Yeah, so the question is about the game publishers. Game publishers are important people in the space. Any big company for that matter isn't going to give you the time of day as a small startup. Which is both good and bad. It means you don't need to talk to them because they're not interested in you. But it means you actually just can't talk to them. We tried but no one wanted to talk to us. They did once we started getting some traction and becoming a bit of a player in the space.

I don't really want to talk that bad about them because they were nice enough about it. When you are a tiny little startup, there are lots of tiny little startups, and they don't have the time to talk to all of you. As we've gotten bigger, game publishers have become increasingly important for us. If I was to talk about who Twitch does user interviews with now, who we pull information form now, it would include game publishers. Definitely! They've become much more active in the space. They weren't particularly active three or four years ago. The really important user interviews in general are from the pool of people you care about, and that is going to shift over time. The people who get you started for the first six months are not who will be using it three years later. It's very important you keep doing this stuff. One of the things that is really easy to do is to do a little bit of it in the beginning and achieve some success and then stop talking to new people. That's a good way to make the next set of features you build be not as good as the first ones.

Audience Member 10: How do you give good user feedback if you're a user?

Emmett Shear: How do you give good user feedback? I want a user to tell me what they are really thinking. What their problems really are. To just sort of ramble. I want someone to just tell me about stuff in their life. The more you learn about them as a person and the context of what they are doing, the easier it is to understand why they want the things they want. That's really the critical question. What I am looking for in someone when I am doing a user interview is someone who is going to be willing to talk a lot and be willing to give me a full picture. On flip side, if you want to help people out with good user interview feedback, ramble.Just talk about everything.

Alright great. Well thank you very much!

Sam Altman: Thank you very much!

Lecture 17: How to Build Products Users Love, Part II

Hosain Rahman

Thank you Sam for having me. Sam and I have known each other for a long time. We met in the early days when he was on his company journey. He asked me to talk today about the hardware journey of building products. I want to give you guys a little bit of an overview of Jawbone: what we do, how we think about the world, and how that informs how we build products. I'll then go into the process of how we design, how we develop, and how that all comes together through what we do to change categories.

I always like to start with the broadest thinking. The way we look at the world is we think of ourselves at this intersection of really crafted innovation in engineering that's almost invisible to the user in terms of its functionality, even beyond design. We have been designing products for well over a decade now. We think that the conversation has shifted even beyond design into beauty. It's the intersection of engineering meets beauty. The whole point is to help people have a better life with technology.

Largely speaking we play in this world of "The Internet of Things." We were there before there was such a moniker. We have smart devices that have computing power and connectivity with sensors that are measuring all kinds of things. They're wirelessly connected and they're all talking to you. We started on this journey really, really early. Right out of engineering school here, we were developing core technology. We decided to build consumer products around that.

Our first consumer product was the headset. We created a headset that became a wearable computer. It was the first traveling headset. That was when we started thinking about wearable computing. We then invented the wireless speaker space around Bluetooth and audio. I will talk a little about that journey. Most recently, we focused our attention on the wearable health revolution and using a lot of the sensors that we did in the first generation of headsets and applying them to other parts of the body to understand more about users.

Our view of this world, having been here for a long time, is that it is a little bit of a mess. In the internet of things everything is smart, connected, and has an app built for it, but that doesn't mean that it is easy for users. Your microwave, your refrigerator, your car, your Xbox, your Xfinity Comcast, everything has an app. They don't talk to each other. It's really confusing for the user. We think that there is a desperate need for an organizing principal around all of this. This is the core of when we start to think about how we build and opportunities to create products. We think about where the world is going. If there is going to be such a world that everyone's talking about on the internet of things which is happening, you desperately need these organizing principals so that it's easier for users to understand how to come in and interact with these services. So we think that is the shift from less about the actual things to being about the individual user.

When everyone is talking about wearables and you have things like Google Glass and Apple Watch, ultimately what we believe is that when you have things that are on your body 24/7, they become a perfect context engine for everything in the world around you. My phone is not on me. It’s in my jacket or sometimes on the charger. But my Up is on me. And it understands everything that is happening when it's tracking my heart rate. It's tracking my respiration. It's tracking all these different things. When I say context engine, I can tell the smart thermostat, my Nest, that I am hot or cold. That device doesn't have that understanding. I can tell it I am hot but that device doesn't know if I'm hot because I'm sick, I went for a run, or if it's hot outside. I can tell your car that you're falling asleep, agitated, or irritated. This is ultimately where we think the world is moving. Wearables are going to be the center of this revelation around everything being connected and smart. We are going to drive what a lot of those interactions are going to be and how they’re going to work. That's the first principal that we think about. Where are things going? What should we build and how should we think about new categories.

In order for this vision to happen, you actually need to be great at almost everything. We need to be great at what we call the full stack. We have to be amazing at building hardware. These are hardware experiences that people have to keep on all the time. You have to wear them 24/7 because if you don't, then everything that I am talking about is a castle in the air. You can't actually create a service that people engage with or that gets lots of data that can then go power all these other things if it doesn't start with great hardware. So that's where we start, we try to build these magical experiences in hardware that are powered by software. We have developed world class software application expertise. We have to be good there, from an engagement perspective, like an Instagram or WhatsApp.

On the data side we have to know what to do with this massive amount of information. We have to know how to process it, push it, and have it work for the user. We really see ourselves at the intersection of hardware, software, and data. These are three equal stools that have to work together in order to unlock that experience around something that is on you, that knows what's happening and then talks to the rest of world. That's a key piece of what we do. It's different than what a lot of other companies doing. It allows and requires to play at all levels of stack.

This was a complicated thing for us to put together because typically people who are great at hardware understand mechanical engineering, electrical engineering, and how those things interact. They understand how you build tools at scale. They're not typically great at building software and services. It's a very different discipline and requires a very different skill set. When we first put those pieces together, it created a lot of interesting friction in the company. Our software and application team was so used to moving really, really fast and iterating, whereas in the hardware world you've got to take your time because your iteration cycles are much more deliberate. You have tooling that takes sixteen weeks. You can't just tweak stuff and you can't hack it in the same way. It was interesting to see that when we put all these pieces together, the hardware learned to move faster. The software guys thought more about how they could resolve experiences before they actually shipped it, versus just throwing something out and A/B testing it. Then the data science informs all of that with more information to make decisions.

So how do we think about and how do we build products? How do we change categories? First of all, everything for us is a system. We don't think about it discretely as a piece of hardware or discretely as an application or discretely as a platform. We think across the whole thing. This is an example with Up. We have these tracking sensors on the body out rhythms there it connects to the phone where you have this engaging application service experiences. We use the sensors in the phone. That talks to a lot of stuff we are doing in cloud where we are taking all that information, driving insight on it and then we have a huge platform of thousands of developers - where they're thousands of apps that then plugin and also create more experiences. And so we think about it across the whole spectrum. And I, I'll come back to this system think in, in a second.

What does the actual process of creation look like? This is fun for me because we don't actually talk about this very often. We keep it confidential and private and I know that we're on a live cast. It's fun for me to talk about this for the first time. It's a quite deliberate process. This is a little bit of what it looks like. This is a map where we are much unbridled in our imagination in the exploration phase. We start to validate some of our concepts, bring those ideas tighter, and tighten them. And then we actually start to build a product. Launch it and then iterate. That's the simplest way to look at it. I'll take you through each of these steps.

The exploration phase is very wild. It's imaginative. We think about the vision of where the world is going and what our strategy is. What does the brand stand for? You're dreaming. You're imagining it. How do I disrupt? What's the future going to look like? It is a little bit of a science project and we talk about it in that way. We build from inspiration, insight, and raw creativity. We want to create and we try to create a form where that's ok because a lot of times in companies that gets lost. Then we we start to bring in early validation, where I say, " Look everyone, when you're doing this stuff, you have to now take those concepts and prove them like you do in a PhD Thesis." You have your conclusions, you've done your empirical data collection, and you start to say here's where we see it going. Here's what's its going to do. You outline the story.

Once we sign off on that phase, we start to go into a concerting phase. Where we start to really think about the experience and what's possible. This is another interesting opportunity for innovation at a more specific level. How will this come to life and how will we sell that experience? How will we tell that story? Then we decide its program. It goes into a heavy planning phase where we say, "Ok. We're doing this. We've got to ship it. There is no turning back." What are the tradeoffs between all the creativity and all the ideas we want versus what the physics dictates? What are all the different constraints that we have? We start making tradeoffs and we look at how to pull that together.

Then we move into a development phase. It's a hand off between various stages and very functional teams in the company. You pull together and you're solving problems as you implement and launch it. You learn. You see what users think. You start to think about where does this stand in that experience continuum that you've been imaging where the world is going to go. What have we achieved? What haven't we achieved? What have we learned from our users? How does that change what we're thinking? And then we start right over again.

That's the broadest way to think about it. The exploration phase is very much like a building and tinkering process. A lot of it is driven by demo Fridays where people have an opportunity to showcase their work. We find that's a great way to pull it together, pull it into a form where others can consume it and give feedback. It's a really it is a show and tell. Obviously hack-a-thons are a big part of it. There's lots of data that gets driven. It's lead by our Strategic Development Team, which is traditionally called an R&D Team. There is participation from product and engineering, both hardware and software. But they're sort of taking a back seat and they're looking at what these explorations are. The Executives at the company at this phase are more of a signing board. They’re there to poke and prod and tell people, “Hey think about this." Or "Did you try that? How does that work?"

In this phase, in order to move to the next threshold, we think, "Would I give this guy 50 grand?" It's like an angel investment. Would I give this guy 50 grand to go explore this and see if there's is there something to do? And our CTO is the final decision maker. He gets to pick internally and say, "You know what? I like all the feedback. This is the one I want to go chase down and see what happens."

Then we get into the validation phase. This where gets really interesting. It's still led by R&D but they're really poking at the idea and saying, “How does this work? We have leadership meetings with the broader cross functional team. I have to show results. I have to go through a scientific process to outline why this works. Why is it going to happen?" This is when we start formulating an important tool in the company, which is what we call WHYS. Defining the WHY of what we are doing. WHY does this exist? What problem does it solve? I'm going to come back to that in a minute.

At this point it's still an R&D lead, but this is when our industrial design team and a few project guys come in and think, "Ok how can I pull this concept into something physical, if it's hardware? How is that going to interact with the rest of the pieces of the system?" Our product experience team is still driving a lot of the core values and the story boarding, but it starts to become a lot more real, when we start thinking, "Ok, how we will build this? How expensive is this going to be? What's the budget going to be?" At that point, we start to really validate if we can actually build it. Do we have wait three years for batteries to be there? Do we have to wait for this other innovation to happen? Do we have to wait from a budget perspective? Is there a business viability? Then we start to really sketch briefs. This where I come in and make the final decision. "There’s really a there there. And we can now take this to the next level and get into play."

Then we go into the concept phase. This is when the responsibilities shift from the R&D folks to what we call the product experience team. The way we think about product experience at Jawbone is what everyone thinks of as conventional design. So from industrial design to software design to audio design to anything that touches that experience. We have writers on that team. Story tellers. We have ID people like Eve who are Genius creators. We have amazing app level designers, graphic designers, everything. It's all one team and we call that product experience. Their job is to unify us as one organization. That's when they take hold and start to really drive the WHYS. They think about what's possible. There's a lot of innovation and creativity in the actual implementation of how we're going to build and create a product. We start to say what the most important things in that product are. What are the most important problems we're going to solve? We call them "Hero Experiences." What are we going to do? What is the bar that would be acceptable?

At this point we start to really resolve the WHYS, which I will show you again in a minute. Why is this different from the competition? From the category? Where does it go? We don't like to do one off things. We have to see a broader vision. This is part of the creation experience. We look at where do we think the world is moving and think about how this is going to be a stepping stone to that ultimate end vision. That’s where the road map starts to get flushed out. Again, I have the ability here to be the final decision maker with my team and say, "Yup. We're going to move this to this next phase."

Here is also where we look at some of these things. I want to get into some specific examples. We have fast track programs. We took, for example, the Jambox when we were in this phase and we said, “We’re not going to go through another phase. We're going straight into the development process because we want to get this thing out. We want to test it, market it, and move really quickly." So we have the ability to prevent our own process and say let's fast track it. We can recalibrate the go to market possibility.

After this stage it shifts from that Product Experience Team to our Product Managers, who are really defining the business plan. When is it going to launch, when is it going to get into the retail calendar, what is the software release cycle. They are prototyping. They're starting to make a lot of those tradeoffs. "Ok we wanted to build this. We can't do that but here's what we can do. We want to be this way. We want these functional experiences. We are going to sacrifice battery life, whatever it is." That's when we start to really pull those decisions and start to look at it. It's a big juggling act at that point.

The product guys are driving that. That's when again, we look at and synthesize all of what we put together and we say, "Ok. Does it actually cross enough things off our list? Does it meet that minimum viability?" Because we always start with, as you can tell, this very big wish list of what's possible and what we can do. Then we start to wheel it down and ask, "Does this cross enough of the value threshold that we think it's worth pursuing?"

Now can we actually move it into the development phase where again Product Management Team continues to lead it. But now you're starting to really get deep. This is where engineering comes in and is really starting to sign off on building it. Here's the time schedule and how we ship. The Product Team is looking at how we should go deeper. How can we increase engagement? What are the little innovations? What are the tuning? What are the things that we need to do to make all that happen?

We've been fortunate to have a lot of wonderful response to products that we've built. We take a lot of care and time in the details of this development and concerting phase, around little details that create these magical experiences. For example, when you turn on Jambox, you have this pretty cool sound that goes "WOOO." It took months to come up with the right audio tuning. We worked with a lot of different audiographers to create that sound, but every time someone turns it on I see them smile and laugh.

The feel of the rubber retyping: there's one manufacturer in the world that was able to make the rubber at the quality that we wanted and the colors that we wanted for the first Jambox. All of those little magical details. How do you resolve them, even in software? When we had the first UP, when you plugged it in your sleep graph showed up. Even just the animations of how the bars would show up and the way cards would flow, that was a detail that we thought about. How is this going interact, how is the user going to experience it? How are they going to feel it? A lot of that stuff happens even at the stage where you sign off on a program. You're making those kind of decisions all the way through and you're trading off and you're doing it in the context of this bigger picture. Innovation is an opportunity to keep refining and to keep doing all that stuff.

How do we think about it at a broader level? What is the framework for how we think about these user signature experiences? Well we start to think of these WHYS. Which is an articulation of the problem that we are solving. And then the themes around how these become actionable concepts. Then we build these cross functional pods that take a person from the Product Experience Team, a person from Hardware Engineering, a person from Software Engineering, a person from the Data Team, and we put them together. This is the pod that owns that theme or that track and they continue to build that out against the hero features and the inside features.

I am now going to go into more specifics around the WHYS because this is where I spend a lot of my time. Where we are asking a question. It serves as a really interesting framework for us to be able to come back and say "Hey. Did we meet those questions that we asked? Does this thing actually do it?" It also serves as a really good guide post for a lot our creativity and a lot of our innovation so it’s not unbridled. It comes down to a very simple question for us: "What is the user problem that we solve through this experiment?" Whether it's in hardware, software, data, platform, whatever it is, once we solve it, people can't live without it. They may have an absolutely burring need to solve this problem and they can't. Either they are looking for a solution or you never thought you needed it but now you can't live without it. Again, Jambox is great example of that. We talked to some people when were were thinking about making that product.

A little story for you guys, when we launched the Jambox in the fall of 2010, the market for wireless speakers as an overall speaker market was zero percent. Zero percent. Last Christmas, which was Christmas of 2013, it was 78% of the market. In a few years we transformed an industry that has been around since the 50's and 60's, we turned it on its head.

If I had gone out and asked a bunch of people, "Who wants a $199.00 speaker for your mobile phone?" I guarantee you 0% of those people would have said, "I want that thing. I need it and I would be willing to pay for it." But when we did it, it transformed an industry. So this is where these WHYS become super important. Focus in on what you're doing. I'll go through one example in the audio space, which is the Jambox example, and then I'll take you through a little bit of how we did it in UP. Particularly UP24. It starts with what we call category strategy. This is the experience framework.

Our view was that all your content and media experiences are now in your phone. They are no longer in iPads, iPods, or your computer. So we need a different way to interact with it that needs to be as portable on mobile, as high quality. That was our fundamental thinking. Then we said that that experience needs to be seamless across time and space. So you could go anywhere through it, in a different car, traveling, or around the house. That was fundamentally what we were doing. We said that’s why this category should exist. That was the human problem.

Then we said, what's in it for Jawbone? Why should we do this? When you think about the broader macro context, the internet of things, this was our entry into your home. This is why everybody talks about all these things in your house from lights to thermostats to fridges to anything that is connected media is still the killer app that's in your house. It's where we sell millions and millions of units. So we said speakers can be our entry into that world that's around you and it can be thumb of the things that we want to do from a software and service perspective in your home. There's an interesting strategy for solving user problems but then why does it matter to Jawbone. Those two have to go together because A) we're not a philanthropic not-for profit-industry and B) if you do this well, it allows you to keep making great products, to keep moving forward, and to keep doing interesting things. That's how we put that together.

Then we built what we call the experience conium. Is where it is today? And when we started it was a Blue Tooth speaker. Right? That was the core enabling technology that it allows us to connect to stuff. Where do we think it's going to go tomorrow? What happens when we can dream in future? We start to really try to live into tomorrow and the future and sort of the thing that we built today as a gradual - stepping stone to graduate users starting one place continue to move - continue to move through that. That gives us a view of how we also make these tradeoffs. Because we said we're not going to put this into this product but we have a space for it in the next one. We know that we can move users to that and they will be ready for it. That’s a lot of where we build stuff is we sort of define that experience contain.

We talk about this a lot. We don't think of ourselves as a hardware team or a software team or a data company. We think of ourselves as an experiences company. It's not just about this physical device or that feature. It's about the system. It's about how the pieces come together. So when we start to define these WHYS, they become the problem statement. We say, "Ok how do we use a piece of hardware? How do we use a service in the cloud? How do we use an application? A sound? A button? How do we solve this user experience problem that we have and what’s the right distribution across that system? Where should you attack the problem? Where do we need to innovate and where do we need to pull it together" That's a big, big, big part of the thinking that helps us in doing.

When we think about these experiences, it's really about the context of why it's magical to the user. Like I said, the system is a flagship and then it has to go to a level of emotional connection where you feel that without it you're lost. I am going to go home and get it if I don't have it. Those are the principals that govern all these things. We have to keep asking ourselves those questions. Is it doing that? We pull this all together to create an experience framework. This is essentially a brief for your engineering team. for your design team, and they can go back and say, "What are we doing and why are we doing it? How does that work? How do we create it?"

And then we have a whole process, Blueberry is one of the internal code names. But the user experience process starts with a better resource, so we do actually listen to users and talk to them. But we talk to them in a very specific way, we start looking for those key insights. We concept them and then we start to build. This is why we go to find those lists of consumer problems. The principles, how do we think about approaching that? What are the solutions?And then what's required in the product to make that happen. Sam?

Sam Altman: Could, could you talk about how you balance the fact that a user would never, told you they wanted to pay $200 for a wireless speaker.

Hosain Rahman: Yeah. With user research in front of this process. There's a lot of layers to user research. That's a great question. You guys probably aren't familiar enough with this yet, but there are standard tools for what people do in focus grouping, where they say, "Would you try this, would you pay this, do you want this feature, what do you care about." That's one way to do it. We don't usually get really great answers. We ask different kinds of questions. We say, "How much music do you listen to when you are with other people? How do you play that music? Do you listen to a headphones, or do you listen over the speakers on your phone? How often are you with other people? How often do you want a personalized experience? How often do you want to share? How often do you?"

We ask a lot of questions. We just ask different ones. We don't ask them specific things about, do you want this or do you want that. We ask them: how do they behave? How do they live? A great example is the iPod. If you said to somebody, "If you could put a thousand songs in your pocket and take them anywhere," that's cool. Not, "Do you want a digital portable music player?"

Again it's at a price that was more than your phone. So you have to separate what are questions that you can ask that are going to help make you smarter about your thesis versus trying to get somebody to validate it for you. That's the real separation. No one's going to tell you what to build, if they do then they should do it and not you. You're the one who's making that decision, you've got the thesis, you've got the creative idea, you've got the innovation. You gotta use these people to help you make it better and to refine your thinking. That's the difference. Make sense?

I'm going to switch over to Up24, which is the product that we've had on the market, our wireless products for health tracking. The WHYS of Up24 are really simple. First of all, let me start with the WHYS for Up. The idea there was there's so much that we know about the world today, through Twitter, Facebook, social media, access to the internet, Google, etc, but we know nothing about ourselves. We have no idea why some days I sleep eight hours and feel terrible, but some days I sleep three and feel awesome.

Our thought was: could we take a lot of this sensor technology, help people understand more about themselves, and start to then make better decisions about how they live better? That was the first product. This was the second product we said: okay, great, now that we have wireless connectivity, it's not just about Bluetooth or wireless, it's about the fact that I can use that real time flow of information to understand what's happening with me and take action on it. I can get the data in a more meaningful, relevant, contextually important way at the moment that it matters. I can also get back guidance in a structured way that can help me go do things. I want that ongoing encouragement, because everybody knows that they want to be better, but they fall down. They want a fluid way to interact with this.

This is what we're building in Up24. We had this very crisp set of five things that were the WHYS of why we're building this product and why we're doing it. Our point of view was that it was going to, we had this sort of fundamental narrative going back to the experience framework where we said everything we do in UP is about helping people track and understand them, track themselves, was understand, which is taking all that data and converting it into knowledge.

The third part was act, so track, understand, and act. That is our narrative for everything we do in the wearable health space and it will be for the entirety of what we do. It's help people get more information on results. Data is great, understanding is better. Convert that into things that they can create real knowledge that they can then take action on. Anything that we can do to keep the device on, get more information, help them be engaged and then find ways of guiding that behavior was really, really interesting and is the framework for the system.

Then you can start to think about designs how you build your data infrastructure, your insight system, how you process it, how you build the application experience that surfaces it. This is a little bit more of a blowout around track, understand and act. This is the tracking part, which is really fundamentally about the hardware too. It's how do you design the batteries? How do you design the embedded systems and materials? The way it latches on you, how easy is it? So that you create the habit of keeping it on your body. Then you have to take all that data, it's not just visualization of information. If I told you guys' your heart rate was 75, is that good or bad? Who knows the answer to that question? I don't. It depends on what you're doing and who you are and what's happening. Just the data surfacing is not enough. You have to contextualize why that matters, turn it into action. That's the third part. Action is the key. Let me understand the data. Let me understand that when I work out at four o'clock, I get four more hours of deep sleep at night. That's awesome. Let me get a reminder at four o'clock to go work out. That's what we've built. That's a lot of infrastructure to create that experience. That's how we build software. That's how we build hardware. That's how we build sort of the whole system.

Often, we will talk about different kinds of users and what they care about and what we think our userbase is made of. Who's more into weight loss, who wants the social acceptance, who are people who are vain, that just want to look better. There's lots of different things. There are people who have medical reasons for the use of our product. We design different kinds of experiences. We think about using platforms like phones and ways to push notifications as part of the system. We think of notifications as a tool for behavior change. We actually start to go map out these things. What is a smart action? Is it real time, does it feel customizable, does it feel progressive, does it help me, is it really tailored to me? For this particular type of user, we'd go out and storyboard. These storyboards go to our design engineering teams. We work together and they actually start to build off of this. What this does for us is it creates a nice set of constraints. My experience has been constraints are really great because they serve as opportunities to resolve, to refine, to simplify, and push you to find the right answer that will solve the user problem in the simplest way.

We create a lot of those constraints around what we're doing. This is the storyboarding for getting someone to the goal and how they do it and what we use, in real time. Then we put in the secondary experiences, which is if we can do this and we can fit it in, if it's not too cluttered or confusing we'll put it in. That's a little bit of a snapshot into how we build and we've few minutes left so I'd love to answer any questions.

Audience Member 1: So let's say you have a product. You have all these features that you want to create. You're about to enter to the design process. How do you approach the whole problem? How do you break down how it is going to solve the problem? But then, each design feature is not mutually exclusive. How do you approach it holistically? When you have a number of different features and functions that you are trying to build, how do you look at them on a system level rather than in a silo to understand what the tradeoffs are across the entire system?

Hosain Rahman: That's the answer to your question. You do exactly that. You don't think about it in a silo. When it’s a small team, it's really easy because you all are sitting around the table. You're looking at each other. You are making those decisions in real time. As you get bigger, in larger companies, you have to force communication where everyone is in a room and a person says, "If you build it, if you were to constrain me in this way, I can't get the quality spec that you need me to make." And another guy is going to say, "Well if you do that then I can't fit all of the rhythms in at the battery performance that you want."

When you look across the system, everyone has to share what their pains are so you actually understand, "If I make this trade off, it's going to affect me over here." You have to put everyone in a room and start hashing that out. That's what’s on the board and on the walls on everywhere, that's what are we trying to do. Does that trade off still meet it across right across all those, all those different cylos. Because everyone is thinking about the trade off in their bend. They know what they need to accomplish. But again how does that affect the whole, whole thing. We just went through this with UP3 - which is a product we are shipping in a couple of weeks that sort of define the next wave of what happening in the wearable space on the health tracking side. We invented a totally new sensing system. Right? There was RAW science that had been developed that `we productive really fast and even just trade off on what the electro materials were. How it affect so n reliability. Source sing. You know, signal performance. And just - these guys weren't talking their way to get in a room. Do daily calls for three hours where they are going through each of their thing. It's tedious. But we're figuring out and we're knock it down. So it, its - when you're small it's real easy you just draw and look at it. But you have to always have that definition of what you trying to do across the system. That's why a lot of what I was talking about was a much higher level. What problem we're solving. Where does it go? And how all of these pieces are formed.

Audience Member 2: Should we start focusing on one small thing or should it focus on the system itself?

Hosain Rahman: A system is a mindset. It's not actually a system. There are simple systems. There are complex ones. A plane is a very complex system. A car is a very complex system. There are other products we make that are much simpler. A phone is a complex system. An application you should think of as a system. Storage. The front end experience. What you're doing is connecting. That’s all a system. So that’s more what of what I mean about systems. For us, a system is hardware, software, and data but I think within anything there's always a system. It’s more just thinking about how the tradeoffs work across all the different pieces that work together.

Audience Member #3: What’s the decision making process between making unrelated products and saving space for fitness tracking, for different versions of or Jambox. What goes into that?

Hosain Rahman: We have a grand unified theory about how these experiences come together. What happens, it touches a little bit on the context engine, when you have things on your body that can make everything in the world around you smarter. If I know the emotional state of a user, I can tell Spotify what song it should play on the Jambox. I can tell the TV that you didn't like that commercial and they should fast forward to the next one. Or I can tell you to not watch Game of Thrones on a Sunday night because you don't sleep well.

I'm serious. These pieces go together. We do think at that level. We start to say, "What are the building blocks to get there? And how do we establish credibility? How do we establish a distributing system? How do we establish manufacturing scale? How do these pieces come together?"

Lecture 18: Mechanics--Legal, Finance, HR, etc.

Kirsty Nathoo

Sam Altman: Kirsty and Carolynn are going to talk about Finance and Legal Mechanics for Startups. This is certainly not the most exciting of the classes, but if you get this right, this is probably the class that helps you avoid the most pain.

Thank you very much for coming.

Carolynn: Like Sam said, this lecture is about the Mechanics of the Startup. Kirsty and I are going to be talking about the basic legal and accounting issues that your startup may face in the very beginning.

I was watching Paul Graham's video and at one point he says, "Founders don't need to know the mechanics of starting a startup." And I thought, "Oh no! That's exactly what Sam titled this lecture."

What PG actually says is that founders don't need to know the mechanics in detail. It's very dangerous for founders to get bogged down in the details. That's exactly right. Kirsty and I can't give you the details in forty-five minutes anyway.

Our goal here today is to make sure that you do know better than to form your startup as a Florida, LLC.

Kirsty: As Sam mentioned, we were also worrying that this was going to be pretty boring for you to listen to an Accountant and a Lawyer talking. You've had some really amazing founders talking about really interesting things. But like Sam said, if you know the basics, you can get yourself set up in the right way, avoid pain, stop worrying about it, and then concentrate on what you actually want to do, which is make your company a success.

We refer to this term "startup" all the time. In the back of your head, you probably know a "startup" has to be a separate legal entity. We're going to talk a bit more about how you actually set that up and what that means.

You also probably know that a startup will have assets, IP, inventions, other things, and that the company needs to protect those. So we'll talk a bit more about that and about raising money, hiring employees, and entering into contracts.

There are a few other things that you need to talk about when setting up your company which ferret out a few issues amongst founders. Who's going to be in charge? How much equity is everybody going to own?

Carolynn: The first thing we're going to talk about is formation. Your startup is going to be a separate legal entity. You probably already know this but the primary purpose for forming a separate legal entity is to protect yourselves from personal liability. If your company ever gets sued, it's not your money in your bank account that the person can take. It's the corporation's.

Then the question is: where do you form one? Theoretically you have fifty choices, but the easiest place is Delaware. I'm sure you're all familiar with that as well. Delaware is in the business of forming corporations. The law there is very clear and very settled. It's the standard. The other thing is that investors are very comfortable with Delaware. They already invest in companies that are Delaware corporations. Most of their investments are probably Delaware corporations. So if you are also a Delaware corporation then everything becomes much simpler. There's less diligence for the investor to do. You don't have to have a conversation about whether or not to reincorporate your Washington into Delaware.

We had a company at YC about two years ago that was originally formed as an LLC in, I'll say Connecticut. The founders had lawyer friends there who said that this was right way to do it. When they came to YC we said, you need to convert to Delaware. The Lawyers in Connecticut did the conversion paperwork and unfortunately they didn't do it right. They made a very simple mistake, but it was a very crucial mistake. The company was recently raising money, a lot of money, and this mistake was uncovered. The company thought it was a Delaware corporation for a couple of years but in fact it was still a Connecticut LLC. I'll just say this: four different law firms were needed to figure that one out. Two in Delaware and one in Connecticut. One here in Silicon Valley. The bill right now is at five hundred thousand dollars for a conversion mistake.

What's the take away here? Pretty simple. Keep it really simple and familiar for yourself. The reason we incorporate all companies the same way at Y Combinator is because it's easy. Don`t get fancy. Save yourself time and money.

Kirsty: Once you decide that you're going to be a Delaware corporation, how do you actually set that up? It requires a few different steps. The first one is really easy. You literally just fax two pieces of paper into Delaware saying we're going to set up a corporation. All that does though is create a shell of a company. It doesn't actually do anything within the company. After that, you then need to complete a set of documents that approve the by-laws of the company. It creates a board of directors. It creates officers of the company. Delaware requires that someone has the title of CEO, President, and Secretary.

At this point, you also need to complete documents that assign any inventions or any code that you as an individual create so that the company actually owns that. Remember, at this point it's a really good thing to think about, "Am I doing this as an individual, or am I doing this on behalf of the company, which is a separate entity." You have to maintain that split in your mind.

There are services that can help you get incorporated. You can use a law firm, but there are also other online services that help. The one that we often use with YC companies is called Clerky. They are set up so that all that standard basic documents are used and they get you set up in a very vanilla way so that you can move on and keep focusing on what you need to do.

A note on paperwork. You're creating documents. These are really important documents that are going to be setting what the company does and what the company is. It's really important that you actually keep these signed documents in a safe place. It sounds so basic but we get so many founders saying, "Oh, these are just some documents." They have no idea what they are or where there are. So really, really make sure that you keep them in a safe place. Let's be honest, filing documents is not the glamorous part of running a startup. The times where this is crucial are going to be high stress times in the startup's life. It's likely when the company is raising a big Series A or if the company is being acquired. The company will have to go through due diligence and there will be lawyers asking for all this stuff. If you don't have it and you don't know where it is, you're making a stressful situation even more stressful.

The key thing here is keep those documents in a safe place. Keep them organized. It will make your life so much easier.

Carolynn: Now we're going to talk about equity. We're going to touch on a couple different things in this section. The first thing that we're going to talk about is equity allocation. If your company stock is high, how to divide the pie. You have to talk about this with your co-founders. Why is this important? If you're a solo-founder this isn't important. If you are a team of two or more, then this issue is absolutely critical.

The first thing that you need to know is that execution has greater value than the idea. What do I mean by that? A lot of Founder Teams give way too much credit and therefore a lot of the company's equity to the person who came up with the idea for the company. Ideas are obviously very important but they have zero value. Who's ever heard of a billion dollar payment for just an idea? Value is created when the whole Founder team works together to execute on an idea. You need to resist the urge to give a disproportionate amount of stock to the Founder who is credited with coming up with the idea for the company.

The next thing you want to think about is if the stock should be allocated equally among the founders. From our perspective the simple answer is probably yes. Stock allocation doesn't have to be exactly equal, but if it's very disproportionate, that's a huge read flag for us. We wonder what conversation is not happening among the Founder team when the ownership isn't equal. For example, is one Founder secretly thinking that this whole startup thing is temporary? Has one Founder overinflated the work that he or she has already done on the company? Or overinflated his or her education or prior experience? Do the founders really trust each other? Have they been honest with each other about their exceptions for the startup and for the future? When ownership is disproportionate, we worry that the founders are not in sync with one another.

Thirdly, it's really important to look forward in the startup. Said another way, all the founders have to be in it one hundred percent. Are they all in it for the long haul? If the expectation at your startup is that each Founder is in it one hundred percent, for the long haul, then everything that happened before the formation of the company shouldn't matter. It doesn't matter who thought of the idea, who did the coding, who built the prototype, or which one has an MBA. It will feel better to the whole team if the allocation is equal because the whole team is necessary for execution. The take away on this point: in the top YC companies, which we call those with the highest valuations, there are zero instances where the founders have a significantly disproportionate equity split.

Kirsty: You've had the conversation about to split the equity but then what? We talk to many founders who are actually surprised that they have do something in order to own this stock. They think that talking about it is actually enough. This is another situation where you have to think about you as an individual versus you as a representative of the company. And if you equate this to a large company, if you worked at Google and were told that, as part of your compensation package, you would be receiving shares, you would expect to sign something to get those shares. If you didn't, you would be thinking, "What's going on here?" It's the same thing with a small company.

In this case the document that you're signing is a stock purchase engagement. You as an individual buy the shares from the company. In any situation, if you're buying something there's a two way transaction. In this case you're getting shares in return for either a cash payment or for contributing IP or inventions or code to the company so that the company actually owns everything that you've done in the past. We also refer to that stock as being restricted because it vests over time. We're going to cover that next, in more detail. As a result of the stock being restricted and vesting, there's one very crucial piece of paper that we talk about until we're blue in the face to everybody because there's actually no way to go back and fix this. This is one of the things that has blown up deals in the past. We've seen companies who haven't filed what's called an 83(b) Election, and deals have blown up. I'm not going to go into detail about the 83(b) election, but it affects your individual taxes and it affects the company’s taxes. It can have a big impact. The main things here are sign the paper work, sign the Stock Purchase Agreements, sign the 83(b) Election, and make sure that you actually have proof that you sent that in. If you don't have the proof it just goes into a black hole at the IRS. Investors and acquirers will walk away from a deal if you can't prove that.

Carolynn: The next thing we are going talk about is vesting. I imagine that many of you are familiar with vesting, but just in case, vesting means that you get full ownership of your stock over a specific period of time. We're talking about the stock that Kirsty just said. You bought your stock of your company and you own it and you get to vote, but if you leave before this vesting period is over, then the company can get those unvested shares back. When your hear restricted stock, it means that the stock is subject to vesting. The IRS speak for this is, "Shares that are subject to forfeiture."

What should a typical vesting period be? In Silicon Valley the so called standard vesting period is four years with a one year cliff. This means that after one year, the Founder vests in or fully owns twenty-five percent of the shares. Then the remaining shares vest monthly over the next three years. Here's an example. Founder buys stock on Christmas day, let's say, and then quits the company on the following Thanksgiving. So before the year has passed. In that case the Founder leaves with zero shares, because the cliff period hasn't been met. If the Founder quits the day after the next Christmas, so a year and day later, he or she is vesting in twenty-five percent of the shares. In that case the one year cliff has been met.

What happens to the shares when a Founder stops working at the company? The company can repurchase those shares. In the example I just gave where the Founder quit a year and a day after purchasing the shares, seventy-five percent of the shares are still invested and the company will repurchase that full seventy-five percent of the shares from the Founder. How? They just write the Founder a check. That's how the Founder brought it. It's the same price per share that the Founder paid, so it's simply giving the Founder his or her money back.

So why have vesting? Why would founders do this to themselves?They're doing this to their own shares. The number one reason why vesting is important has to do with founders leaving the company. If you didn't have vesting and a founder leaves, a huge chunk of the equity ownership leaves with or her. Obviously that is not fair to the founders left behind. We're actually going to talk about this a little bit more when we get to the the "Founder Employment" slide. I will go into that in more detail.

The other reason to have vesting is the concept of skin in the game, the idea that founders need to be incentivized to keep working on their startup. If the Founder can walk away with his or her full ownership at any point and time, then why would you stay and grind away? Startups are hard.

Do solo founders need vesting? They do and the reason is because the skin of the game concept applies to solo founders as well. Investors want to see all founders, even solo founders, incentivized to stay with at the company for a long time. The other reason that solo founders should put vesting on their shares is to set an example for employees. You can imagine it would be inappropriate for a Founder to tell an employee that he or she has to have four year vesting on his or her shares but the founder doesn't think he or she needs any on their own shares. It's a culture point. A founder who has vesting on his or her shares then sets the tone for the company by saying, "We're all in it for the long hall. We are all vesting on our shares. We're doing this together."

Vesting aligns incentives among the founders if they all have to stick it out and grow the company before they get any of that company. Investors don't want put money in a company where the founders can quit whenever they feel like it and still have a big equity ownership stake in that company.

Kirsty: Moving on. We've now got a beautifully formed corporation in Delaware. Everyone has their stock. It's all in the plain vanilla standard paperwork. Then what? The next stage of a company's life is to raise money. We know that you already heard a lot from a lot investors and from founders in this set of classes. They've been talking about the tactics of how to raise money but what about the paperwork? What about when somebody actually agrees to invest?

In terms of logistics, in very simple terms there are two ways to raise money. Either the price is set for the money that comes in or the price isn't set. By price we mean the valuation of the company. Rounds can actually be called anything. People can name them whatever they want, but generally when you hear the term seed round, it mean that the price has not been set. Anything that's a Series A or Series B is something where the price has been set.

Not setting the price is the most straight forward, fast route to getting money. The way that this is done is through convertible notes or safes. Again, this is a two way transaction. It's a piece of paper that says, for example, that an investor is paying one hundred thousand dollars now and in return has the right to receive stock at a future date when the price is set by investors in a priced round. It’s important to note that at the time that paperwork is set, that investor is not a shareholder and therefore doesn't have any voting rights in the company. They will have other rights which Carolynn is going to talk about separately.

Of course investors want something in return for putting in money at the earliest, riskiest stage of the company's life. This is where the concept of a valuation cap comes in, which I'm sure many of you heard mentioned before. The documents for an unpriced round set a cap for the conversion into shares that's not the current valuation of the company. It's an upper bound on the valuation used in future to calculate how many shares that investor is going to get. For example, take an investor that invests one hundred thousand dollars on a safe with a five million dollar cap. If a year later the company raises a priced round with a valuation of twenty million dollars, then the early investor would have a much lower price per share. About a quarter. Therefore their hundred thousand dollars would buy them approximately four times more shares than an investor that was coming and putting in a hundred thousand in that Series A priced round. That's where they get their reward for being in early.

Again, this is a situation where you need to make sure you have the signed documents. Services like Clerky can help with that. They have very standard documents that most of our YC companies use to raise money.

A couple of other things to think about when you are raising money, um. Hopefully you got a really hot company that, that's doing great and it's really easy to raise money. But you should be aware that all these people throwing money at you does have some down sides. Um, so the first thing is to understand your future dilution. So, if you raise, let's say two million dollars on safes with a valuation cap of six million dollars, then when those safes convert into equity, those early investors are going own about twenty-five percent of the company. And that's going to be an addition to the investors that are coming in at that priced round who may want to own twenty percent of the company. So you're already at that point given away forty-five percent of the company. So is this really what you want? And you know the answer might be yes. Um, remember that some money on a low valuation cap is infinitely better than no money at all. And if those term that you can get then, then take that money. Um, but it's just something to be aware of and to follow through the whole process so that you can see where this is going to lead you down the road.

The other thing to keep in mind is that investors should be sophisticated. They have enough money to be able to invest. They understand that investing in startups is a risky business. We see so many companies say, "Oh yeah. My uncle put money in or my neighbor put money in." They've put in five or ten thousand dollars each. Often those are the investors that cause the most problems going forward because they don't understand how this is a long term gain. They get to the point where they're sitting thinking, "Hmm. I could actually do with that money back because I need a new kitchen." Or, "This startup investing is not actually as exciting as all the TV shows and movies made it out to be." That causes problems for the company. They're asking for their money back. Be aware that you should be raising money from people who are sophisticated and know what they're doing. The term that you'll hear that refers to these people are that they are accredited investors.

The main point here is keep it simple. Raise your money using standard documents. Make sure that you have people who understand what they are getting into and understand what you're getting into in terms of future dilution.

Carolynn: Ok. You're raising money. You understand what you're selling. You figured out the price. You got down the logistics that Kirsty just described. What you may find is that you don't understand some of the terms and terminology that your investors are using. This is ok, but you do have a burden to go and figure it out. Don't assume that just because you have agreed on the valuation of the price, that all the other stuff doesn't matter. It does matter. You need to know how these terms are going to impact your company in the long run. At Y Combinator, Kirsty and I hear founders say all the time "I didn't know what that was. I didn't know what I was signing. I didn't know I agreed to that!" The burden is on you to figure this stuff out.

We're going to go over four common investor requests. The first one is a board seat. Some investors will ask for seat on your company’s board of directors. The investor usually wants to be a director either because he or she wants to keep tabs on their money or because he or she really thinks they can help you run your business. You have to be careful about adding an investor to your board. In most cases you want to say no. Otherwise make sure it's a person who is really going to add value. Having money is very valuable but someone who helps with strategy and direction is priceless. So choose wisely.

The other things is advisers. They are so many people who want to give advice to startups. Few people actually give good advice. Once an investor has given your company money, that person should be a de facto adviser but without any official title and more importantly without the company having to give anything extra in return for the advice.

At Y Combinator, we've noticed that whenever a startup manages to garner a celebrity investor, the celebrity almost always asks to be an adviser. We have a company that provides on-demand bodyguard services. An NBA basketball player invested and asked to be an adviser and then asked to be given shares of common stock in exchange for adviser services. The service that this person had in mind was to introduce this company to other professional basketball players who might want to use an on-demand bodyguard. This celebrity just made a big investment, shouldn't he want to help the company succeed? Why does he need something extra? All investors who can help should do so. Asking for additional shares is just an investor looking for a freebie.

Next we're going to talk about pro-rata rights. Very simply, pro-rata rights are the right to maintain your percentage ownership in a company by buying more shares in the company in the future. Pro-rata rights are a way to avoid dilution. Dilution in this context means owning less and less of the company each time the company sells more stock to other investors.

This is a really basic example, but say an early investor buys shares of preferred stock and ends up owning three percent of the company once the financing has closed. The company raises another round of financing. The company will go to this investor who negotiated pro-rata rights and say, “Hey. We're raising more money. You're welcome to buy this many shares in the new round to keep your ownership at approximately three percent."

Pro-rata rights are a very common request from investors. They are not necessarily a bad thing, but as a founder you absolutely need to know how pro-rata rights work. Especially because the corollary to an investor having pro-rata rights to avoid dilution is that founders typically suffer greater dilution.

The final thing is information rights. Investors almost always want contractual information rights to get certain information about your company. Giving periodic information and status updates is not a bad thing. At YC we encourage companies to give monthly updates to their investors because it's a great opportunity to ask for help with introductions or help with hiring. That kind of thing. You have to be really careful about overreach. Any investor saying they want a monthly budget or weekly update, that's not ok.

The takeaway here is that just because the type of financing and the valuation has been negotiated doesn't mean that everything else is unimportant. You need to know everything about your financing.

Kirsty: Moving on to after you've raised money. The company bank account probably has more zeros in it then you've ever seen in your life. Then what? This is where you actually start incurring business expenses. Business expenses are the cost of carrying out your business. Paying employees, paying rent for an office, hosting costs, the cost of acquiring customers, that kind of thing. Business expenses are important because they get deducted on the company's tax return to offset any revenues that are made to lower the taxes that the company pays. On the flip side, if the company incurs a a non-business expense that is not deductible on the tax return, that can increase the profits of the company that have to pay tax on them.

Again, this is a separation issue. The company will have its own bank account, out of which the company's expenses should be paid. Um, again think about this from a, from a large company, if you were working at Google, you would not use a Google credit card to buy a tooth brush and tooth paste.

Remember that the investors gave you this money. They trusted you with a huge amount of money. They want you to use that money to make the company a success. It's not your money for you to spend how you please. Believe me, we've had some horror stories of founders who've take that approach. We had one founder who took investor money and went to Vegas. By his Facebook photos, boy did he have a good time. Needless to say he's no longer with the company. This is stealing from investors.

The concept of business expenses can get a little bit blurry, especially in the early days when you're working in your apartment twenty four hours a day. The way to think about it is, "If an investor asked me what I'd spent their money on and I had to give them a line by line break down, would I be embarrassed about any of those lines?" If you would, it's probably not a business expense.

The other thing to bear in mind is that you're busy running your company at ninety miles an hour, so you don't have to necessarily think about the book keeping and accounting at that point. However, it's crucial to keep the receipts so when you do engage a book keeper or a CPA to prepare your tax returns, they can figure out what are business expenses and what aren't business expenses. They're going need your help as a founder to do this. The way make your involvement as small as possible is to keep those documents in a safe place, so you can easily refer to them.

If you remember nothing else, do not go to Vegas on investors' money. Spend that money wisely.

Carolynn: In this section we're going to hit a couple topics in this section. The first one is "Founder Employment." Why are we talking about founder employment? As we said already, the company is separate legal entity. It exists completely separately from you as founders. As prestigious as we think the title founder is, you're really just a company employee and founders have to be paid. Working for free is against the law and founders should not let their company take on this liability. You wouldn't work for free anywhere else, so why is your startup an exception? Companies have to pay payroll taxes. We had a YC company that completely blocked their payroll taxes for three years. It was huge expensive disaster and in extreme cases, people can actually go to jail for that. Fortunately not in this case, but it's bad. The moral of this story is set up a payroll service. This is something that is worth spending your money on. Don't go overboard on lavish salaries. Minimum wage. This is still a startup and you have to run lean.

Now I am going to mention founder break ups. First, what is a founder break up? In this context, I'm talking about one founder on the team being asked to leave the company. founders are employees, so that means your co-founders are firing you. Why are we talking about break ups in founder compensation? At YC we have seen a ton of founder break ups and we know that the break ups get extra ugly when the founders haven't paid themselves. Why? Unpaid wages become leverage for the fired founder to get something that he or she wants from the company. Typically that is vesting acceleration. The fired founder says, "Hey. My lawyer says you broke the law by not paying me. If you pay me and you give me some shares that I am actually not really entitled to, I'll sign a release and make all this ugliness go away." If you're the remaining co-founders, you're probably like, "Sounds like a good deal." Now you have a disgruntled person who owns a piece of your company and, even worse, the remaining workers are working for that ex-founder. They are building all the value in the company and the ex-founder who got fired is sitting there with all their shares going, "That's right. Make it valuable."

The take away here is avoid problems by paying yourself. Paying your payroll taxes and thinking of your co-founders wages like a marital pre-nup.

Kirsty: As the founders, you are going to need to hire employees. Much has been said in previous classes about how to find those people, about what makes a good fit, and about how to make them really productive employees. When you actually find somebody, how do you hire them? What's involved? Employment is governed by a huge raft of laws. Therefore, it's important to get this right. It's again the nitty gritty stuff that, as long as you know the basics, you can probably keep yourself out of most situations. As soon as things get complicated, you need to get yourself involved with a specialist.

the first thing you need to do is figure out if the person is an employee or a contractor. There are subtle differences to this classification. this is important to get right because the IRS takes a big interest in this. If they think you got it wrong they will come after you with fines.

Both an employee and a contractor will require documents that assign any IP that they create to the company. That's obviously really important. The form of the document is very different for each type of person and the method of payments are very different. Generally a contractor will be able to set their own work hours and location and they will be given a project where there is an end result. How they actually get to that will not be set. They'll be using their own equipment and they won't really have any say in the day to day running of the company or the strategy going forward. A contractor will sign a consulting agreement. When the company pays them, the company doesn't hold any taxes on their behalf. That responsibility is on the individual. At the end of the year, the company will provide what's called a form 1099 to the individual and to the IRS, which they use to prepare their personal tax returns.

The opposite side of this is an employee. An employee will also sign some form of IP assignment agreement, but when the company pays them, the company will withhold taxes from their salary. The company is responsible for paying those taxes to the relevant state or federal authorities. At the end of the year the employee receives a W2 form, which will then be used to prepare their personal tax returns.

The founders need to be paid. So do employees. It isn't enough to just say, "Well, I am paying them in stock. That can be their compensation." They need to be paid at least minimum wage. In San Francisco, which has a slightly higher minimum wage than California as a whole, that works out to about two thousand dollars a month. It's not a huge amount but it can add up.

There's are other things that you need to make sure you have if you have employees. The first thing is that you're required to have workers compensation insurance, especially if you're in New York. The New York authorities that look after this will send really threatening letters saying, "You owe fifty thousand dollars in fines because your one employee that's being paid minimum wage has not paid the twenty dollars a month of workers compensation fees." It is really important that you set that up. The other thing that is important is that you need to see proof that the employee is authorized to work in the US.

Founders are not payroll experts and nobody expects you to be one. This is all about the basics. You absolutely must use a payroll service provider who will look after this for you. Services like Zen Payroll are focused on startups. They help you get this set up in the easiest way possible so you can go back and concentrate on what you do best. In the example that Carolyn gave just a few minutes ago, if that company had actually set themselves up with a payroll service provider, all of that heartache would have gone away because it would have been looked after for them. They were trying to save money by not doing it and look where it got them.

That's the key thing. Use a payroll service provider and make sure that you understand the basics of employment.

Carolynn: Somebody at YC once said, "You're not a real founder until you've had to fire somebody." Why is that? Because firing people is really hard. It's hard for a lot of reasons, including that founders tend to hire their friends. They tend to hire former co-workers or they get close to their employees because working at a startup is really intense. But in every company there's going to be an employee that doesn't work out and firing this employee makes a founder a real professional because he or she has to do what is right for the company instead of what is easy. Best practices for how to fire someone: number one, fire quickly. Don't let a bad employee linger. It's so easy to put off a difficult conversation but there is only downside to procrastination. If a toxic employee stays around too long, good employees may quit. If the employee is actually screwing up the job, you may lose business or users.

Number two , communicate effectively. Don't rationalize. Don't make excuses. Don't equivocate about why you are firing the employee. Make clear direct statements. Don't apologize. "We're letting you go," not, "I'm so sorry your sales didn't take off this quarter, blah, blah, blah." Fire the employee face to face and ideally with a third party present.

Number three, pay all wages and accrued vacation immediately. This is a legal requirement that we don't debate or negotiate.

Number four, cut off access to digital systems. Once an employee is out the door, cut off physical and digital access. Control information in the cloud. Change passwords. We had a situation at YC where one founder had access to the company's GitHub account and held the password hostage when his co-founders try to fire him.

Number five, if the terminated employee has any invested shares, the company should repurchase them right away. The takeaway here is that, surprising as this may sound, one of the hallmarks of a really effective founder is how well he or she handles employee terminations.

Kirsty: The basic tenant to all of this is keep it simple. Do all the standard stuff and keep it organized. Make sure you know what you're doing. Equity ownership is really important, so make sure you are thinking about the future rather than the three months of the history of the company. Stock doesn't buy itself, so make sure you do the paperwork.

Make sure you actually know about the financing documents that you're signing. It's not enough to just say, "I'll take your hundred K." You and the employees need to be paid. Everybody needs to assign IP to the company. If the company does not own that IP, there is no value in the company. If an employee must be fired, do it quickly and professionally.

We didn't mention knowing your key metrics. At any time you should know the cash position, you should know your burn rate, you should know when that cash is going to run out so you can talk to your investors. A lot of running a startup is following the rules and taking it seriously. It's not all the glamorous bits that we see in movies and TV shows. You do have to take that seriously.

Audience Member #1: How would you advise searching for an accountant and when in the process do you need one?

Kirsty: There are two different things. There is a book keeper and there is a CPA, an accountant. Generally book keepers will categorize all your expenses and CPAs will prepare your tax returns. In the very early days it's probably fine for the founders to just be able to see the bank statements and see those expenses coming out. Tax returns have to be prepared annually, so at some point in that first year of the company's life, some service is going to need to be engaged to do that. It's not worth the founders time to do it. There are services available like inDinero which try to make things as effortless as possible from the founders's point of view. You do need to get a CPA at some point because you need to file your annual tax returns for the company.

Audience Member #1: How do you find one?

Kirsty: Finding one is tough. The best is through recommendations. With any kind of specialist, a CPA or an accountant or a lawyer, it's always best to use people who are used to dealing with startups. Not your aunt who lives in Minnesota and doesn't actually know how startups work.

Audience Member #2: All things considered, what should be my budget for incorporating, for the lawyer, for getting the deal to buy for my effort seed rounds? And then for hiring the first employees. How much money should I set aside for that?

Carolynn: In terms of incorporation, don't spend a dime on that. You can do that online. Well, actually it does cost a little bit.

Carolynn: Incorporating online using a service like Clerky is inexpensive. In the hundreds, not in the thousands. You don't need a lawyer for that part. When you actually need to hire a lawyer depends on what business you are starting and how complicated it is. Do you have a lot of privacy policies, is HIPPA involved? You mentioned raising your seed round, how much money are you raising? Who are the investors? What kind of terms are in the term sheet? Sometimes that dictates whether or not you need to get legal counsel.

Kirsty: Services like Clerky can help if you are using very standard documents for the fundraising. There are very basic fundraising documents you can use that cost less than a hundred dollars, which can save you some legal fees.

Audience Member #3: Do you have any advice or comments on the complexity that comes with working with cryptocurrencies or cryptoequities?

Kirsty: Oh wow. That's a tough question to end with. Yes, there are some issues. Often banks struggle to deal with companies that are working with cryptocurrencies because they haven't quite figured out how to deal with that sort of thing yet. Generally a lot of it is very product specific.

Sam: Thank you very much!

Carolynn: You're welcome.

Lecture 19: Sales and Marketing, How to Pitch, and Investor Meeting Roleplaying

Tyler Bosmeny

My name's Tyler. I'm the CEO of Clever. What I want to talk today is about sales. I graduated college, where I studied math and statistics, and thought I was destined for this world of finance. I was about to start at a hedge fund, but at the last second a friend of mine roped me into joining his startup to do sales, which I knew nothing about. I had to figure it out on the fly. I spent a couple of years there figuring out sales for this very early stage company. When it came time to start Clever, I started Clever with two co-founders who were very technical and very product oriented. We wanted to build this product for schools and I thought that experience would have no relevancy whatsoever. It turns out that what I picked up while doing sales at this previous job has been a huge part of what's made Clever grow so quickly today.

A quick background on Clever: we build software for schools. We are an app platform for developers that is used today by about one in five schools in America. We started it about two years ago.

Sales has been key. I want to use this time to share some of the things that have worked for me along the way. Of course, there's a million ways to do this, so you'll find what works for you.

First I want to start about how I used to perceive sales. A lot of people see sales as having mystique around it. It's people who are articulate and impossibly charming. They have these killer closing lines that they use. This is how I saw sales. I think this is how a lot of founders I talk to see sales because they say things to me like, "You know, we're just going to work on the product and build a great product and then when it's finally finished, we're going hire the sales people." What I've learned is that when it comes to "hiring the sales people," as a founder, the reality is that it's you. Paul Graham likes to talk about how there's two things you should be doing at any point in time when you're starting your company. You should be either talking to your users or building your product. The talking to your users part, that's selling. This is intimidating to some people because they're like, "I've never done sales, and I wouldn't even know where to begin." It turns out that as a founder you have some unique advantages that make it possible for you to be really, really good at sales. One of those is your passion for the product and what you're building. The second is your knowledge of the industry and the problem that you're solving. Those two things actually totally trump sales experience from what I've seen.

This is actually my co-founder doing sales. This is what sales looks like in the very early stage of a startup. It's not Don Drapers. It's a lot of calls like these. This is something that even as a founder who has never done it before, is very easy to do but you have to commit yourself. What we did at Clever is we dedicated one founder, which was me, to peel off and say, "Ok, Tyler you gotta go figure this out and work on this full time because it's so important to our business."

The first thing that everybody knows about sales is it's a funnel. You have these different stages of the funnel and you move your customers through it. A pretty common category is the prospecting category. We were trying to figure out who's even interested. Then you're having a lot of conversations, which is the second level of the funnel. Then you're finding out who's really serious and you want to close them and sign the deal. Then you're in the promised land of revenue. I thought it would be interesting to talk about each stage and a couple of strategies that we've used at Clever that have worked well, so that these aren't abstract but hopefully lessons you can use at your start up.

Prospecting is the process of figuring out who will even take your call. There's this guy at Everett Rogers who has created a technology life cycle adoption curve. He describes it as a bell curve where you have innovators who will try new things, early adopters, mid-stage adopters, late adopters, and laggers. One of the things that was really helpful for me in understanding sales at an early start up is he's quantified the tail of this bell curve. This part over here are innovators, those are your potential customers. It might seem discouraging that only 2.5 percent of companies are your potential customers or would even consider buying from a startup that has no users and no revenue, but I found the opposite. I found it extremely helpful to have this frame of mind because when only 2.5 percent of companies will even take your call or consider using your product, you realize what a numbers game this becomes. If you want to reach that 2.5 percent and you want to get some early sales, you're hopefully starting to realize you have to do a lot of calling. You have to talk to a lot of people.

In the early days Clever, this was my job. In the first two months of YC I reached out to over 400 companies trying to to get them to take a call and talk to us about what we were building.

There are three methods that I have found to be most successful in prospecting and getting these people. One is your personal network. That's obvious. I'm not going to spend any time there. Another one is conferences, which is surprising to a lot of people. The one that people are most familiar with is cold email. When I say conferences, people think I am talking about CES or E3. The kind of conferences where sales happen look more like this. In the early days we would go to a lot of these because you've got to go to where your users are. If you're selling to CIO's and there happens to be a gathering of them at a hotel in Milwaukee, guess what? That's where you should be. So we went to conferences like these. We got the attendee list in advance. We'd email every single person in advance and try set up meetings so when we got there every single minute of that trip was was well spent. It was huge in Clever's early days. This is where met all of our earliest customers.

The second thing I mentioned is cold email. A lot of people don't know how to write cold emails. It's actually easy and the key is not to write a lot. Your email should be concise. This is an email template that I used early on. You're welcome to copy it but it's really short. Here's who I am. Here's what I'm building. I'd love to talk to you about this. Could we find time tomorrow? It's really easy and you can customize this for every business you want to sell to. Find out who the right person is to send it to and you can send out quite a few of these.

That's prospecting. The reason this so important is because you have to build that first layer of the funnel.

Then you have to get them to take your call. This is another place where a lot of founders have questions about what to actually do. The biggest thing to take away, in fact if you ONLY take away one thing from this presentation today this should be it, is when you get them on the phone, remember to shut up. That's really surprising to people. So many founders, when I help them with their first sales pitch, would finally get somebody on the phone who wanted to talk to them about their product and they'd be so proud of this thing that they'd been building for the last three months that all they wanted to do was get on the phone and talk about every feature and talk about why it's the greatest thing in the world. I have that temptation too. It's just part of being really proud of something.

It turns out that if you watch the best sales people, the top one percent, or you have a chance to listen in on a call with some of those people, the most surprising thing is how little talking they do. In fact I've seen calls where the sales person told me their goal was to only spend 30 percent of the call talking and have 70 percent of the call be the other person. They would ask a lot of questions. They'd say things like, "Why did you even agree to take my call today?" "This problem that we're talking about solving for you, how do you solve it today?" "What would your ideal solution look like?" They're not doing the talking. They're doing everything they can to find out what this person needs and hopefully understand their problem even better than they do. That's what really great sales is. This is something I drill into everybody at Clever. It's a really important part of sales. If any of you use UberConference, they have this amazing feature where when you hang up a call it sends you an email automatically and tells you how much you talked versus how much the other person talked. Looking at one of those emails, I can tell immediately how likely the sale is based on how much talking we were doing. Do a lot of listening. Really understand their problem.

The other part of this stage that surprises a lot of people is you have to follow up. Here's a lot of different steps that you go through: emailing somebody, not getting a response and emailing them back. Calling them, leaving a voice mail. Having a pricing call. There are probably sixty things on this slide that could be steps for closing a deal. These aren't random things -- this was the second deal Clever ever signed. These are all the different steps that we had to do in order to get this done. You can see there's a lot of really embarrassing things up there. I emailed somebody and they didn't respond. I emailed them again and they didn't respond. I emailed them again. This was from somebody who wanted to buy our product. Isn't that crazy? That surprises a lot of people. I see so many founders who think they have a great call with someone and send an email, but don't hear back. They say, "Oh that person might not be interested." Well guess what? This is what it looks like in the best case. You really have to have kind this unhuman and unreasonable willingness to follow up and drive things to closure.

I qualify with that with one thing which is to say when starting a company your time is extremely valuable because it's your only resource. You couldn't possibly do this for every single person who might buy your product. Your goal should be to get people to a yes or no as quickly as you can. Where you die is if you have a thousand maybes and sometimes I talk to founders who say, "Oh yeah I have this great pipeline of a hundred people who have expressed interest in our product." The maybes are what kill you. If you can get to a yes or a no, in some ways a no is even better than a maybe because it allows you to move on and focus somebody who might be a yes.

So, have a super human level of follow up and ambition, but make sure you're focusing it on the right pieces.

Alright, so you've talked to a ton of people. You've had all these phone calls. You've followed up with them to the point where they know you're not going away and they've got to sign an agreement. This final step is something if you haven't done before it might seem hard but it's actually really simple. It's called red lining. You'll send over an agreement and their lawyers will mark it up. Your lawyers will also mark it up and you kind of go back and forth. If you're part of YC this is really easy because YC has standard template agreements that they give you so you can just use those. But if you weren't part of YC you have to figure this out on your own.

One of the things that I am really excited about is as part of this presentation, YC has agreed to open source their deal documents. The documents that YC founders use are going to be available to everybody. So this should never be a barrier to anyone who wants to do sales for their start up. You've got some great documents. The other place where so many smart people go wrong is they don't remember what their goal is. Your goal is to sign some deals, get some reference customers, get some validation, and get some revenue. If you don't do that, your startup is toast.

In light of that it's really surprising how many smart people will want to do ten rounds of document review over the most minor points because of pride. Whatever. Make sure the agreement is the way you want it but then sign and move on. I've seen founders spend month quibbling over some indemnification clauses. Their business would have been way better off if they'd just signed the deal and moved on to the next one. That's one trap you can fall into.

Another trap that I see founders struggle with a lot is they're talking to a company who says, "I will use your product but I just need one more feature." Or they say, "You know I'd love to use your product but it doesn't have this one feature. So we're just not ready." To most people, especially if you're ambitious, when somebody says that to you, what you want to think is, "Oh. I can build that feature and then they're going to use my product." The problem is it almost never works that way. Somebody telling you that they want to use your product but it's missing this one feature, I would almost map that to a pass in your mind. Nine times out of ten if you actually build that feature and go back to them, there will be one more feature or some other reason that they're not using the product.

If somebody says to you, "There's this one thing that's preventing us from using your product." I would do one of two things. One say, "Well that's great! Let's sign an agreement and we'll put in the agreement that we're going to build this feature." In which case, if you build it you're off to the races. More commonly, what we did at Clever was we would say, "That's great. We're going to wait to see if we hear that demand from more customers." Once you have a lot of customers requesting it, then you should build it. Then you don't have to worry about doing something that's a one off, which is what you really want to avoid.

The other trap I would highly recommend you try to avoid is the free trial trap. The customer says, "Can I get a free trial?" You can't blame them that's a totally reasonable thing to ask for. The problem is when you are starting a startup you need revenue. You need validation. You need users. You need commitment. Free trials get you none of those things. You do all this work and if you end up with a free trial, unfortunately you haven't made as much progress as you think, it's actually terrible. You think you've made progress but at the end of the free trial you're going to have to sell them all over again. The way I handle this that has worked really well is that when somebody says, "Can I get a free trial?" you say, "We don't do free trials. We do annual agreements and what we'll do is for the first 30 or 60 days, if for any reason you're not happy, you can opt out." That's a way to get you the things that you need while giving them the comfort that they might need to take a chance on a startup. That minor change actually makes a night and day difference when you're thinking about these things.

Alright, so you've prospected. You've had a lot of conversations. Now you've closed people. You've gone through the red line process. You worked out the free trials. You're on your way to your first sales. Early on, you can think of sales as just like any other thing at a startup. You don't have to do things at scale. In fact you can purposely do unscalable things to try and get early customers. That's the fun part. The other thing that is important to keep in mind is once you've done this enough, what you should start thinking about is what aspects of this are repeatable. What aspects are we going to scale further? Christoph Janz wrote this really great blog post online about the five ways to build a hundred million dollar company. He talks about how he can have a thousand customers buy a product that costs a hundred thousand dollars. Or he can have ten thousand customers buy a product that cost ten thousand dollars. Or he can have a hundred thousand customers by a product that cost a thousand dollars. Even though you don't need to know on day one which bucket you're going to fall into, most companies do fall into one of these buckets. If you want to be in the elephant category of a hundred thousand dollar product, you're going to have a really high touch sales cycle. That's Salesforce. That's Workday. If you think that you're going to be a rabbit and sell products for a thousand dollars a year and your sales process involves flying out to see them, and eight demos, and three months of redlining, then you probably have to rethink something.

I see a lot of startups who want to be rabbits that don't think about how to do it in a scalable way. That's one area where you can get under water or it just forces you to increase your prices.

This is how I think about different businesses. It will be helpful for you when you get started and once you've done sales to say, "Ok, where am I?" The corollary to that is, "How do I have to price my product to be a viable business?"

Those are some of the things I figured out building sales at a few different companies, specifically on this very narrow stage of zero to one million. After you get to one million, you'll find there's a million blog posts about how to get from five million to fifty million or ten million to a hundred million, but not the zero to one step. I wanted to focus the presentation on that because there's not as much written about it and it is something that I think is very opaque to our founders. I figured this out just by doing it and I'm confident that if you're starting a company you can too. If for whatever reason you would like to join a startup that's figured it out and hone your skills and hone your craft, we are hiring at Clever. That's an option. If you do want start your own company and you have questions about sales, I put my email address up here. Feel free to reach out at any time. I am happy to help.

Thank you.

Sam: Thank you very much! That was awesome! Now we're going talk about a little more about how to raise money. Michael Seibel is first going to talk about how you give a pitch and then Qasar will do investor role playing.

Qasar: Yeah, so this isn't mind blowingly new. It really is a basic blocking attack. And the one point we wanted to make before we get started is we actually don't spend a lot of time at YC focusing on this. The main reason is the best way you can make your pitch better is to improve your company. If you - if you have traction and your product is doing well - these conversations are like the investors want to see you succeed. If you remember anything, it's make your company better and the pitch will be easier.

Sam: We're going to spend the time in three kinds of sections. Before the meeting what Michael will kind of focus on will do kind of a role play what meetings actually look like and then we'll just wrap it up. We are going to do Q&A at the end. We'll save five minutes. If there is something we don't cover please write down your questions and we'll go through them.

Michael: My name is Michael Seibel. I am a current YC partner. I started two companies. One was called Justin.tv, which I ended up selling to Amazon. The other was called Socialcam, which sold to Autodesk. What I want to do is break down and demystify the process of creating a pitch. What happens too often when I see companies coming to talk to me is that they don't know how to simply explain what they do or how to ask for money. That's basically what you have to do as a founder.

We're going to go over four things. The first is your 30 second pitch. You need to be armed with this constantly. This is basically how you talk about your company. It's magic. Whether you're talking to people who want give you money or don't want to give you money, this is your go to.

The second is your two minute pitch. This is for people who are more interested. This is people who you might want to raise money from or people who you might want to get hire. People with whom you need to get a little bit deeper. Notice that's where I stop. A lot of people practice ten/thirty minute pitches or hour pitches. That's all garbage. You can get everything you need done in two minutes. One thing I like to tell founders is the more you talk, the more you have an opportunity to say something that people don't like. Talk less and it will probably be better.

I want to tell you about when to fundraise because I think a lot of companies get this a little bit wrong. And then quickly how to to set up investor meetings.

The 30 second pitch is so simple. It's three sentences. You can take your time. You can breathe when you do this. You don't have to get that much information out. The first is one sentence on what your company does. Everyone I meet for the first time screws this up. You have to be able to do it in a way that is simple and straight forward, that requires no further questioning on my part. You have to assume I know nothing. Literally nothing about anything. This is how you make it super simple. What we tell people is apply the Mom test. If in one sentence you cannot tell your mom what you do, then rework the sentence. There is a one sentence explanation that your mom or your dad is going to understand. So really, really start there. It's ok if you use basic language. It's ok if you say, "Hey we're Airbnb and we allow you to rent out the extra room in your house." That's simple! You don't have to say, "We're Airbnb and we're a marketplace for space." I don't know what that is! That's going to require more time. Use simple language, it's very important.

The second is in a multi-billion dollar market, it's pretty simple to do this. You know Airbnb might say, "How big is the hotel market? How big is the vacation rental market? How big is the online hotel booking market?" These are simple numbers to look up on Google. It makes an investor understand, "Oh wait. If we're big, if we really blow this company up, it could be worth billions of dollars." Don't skip this up. Second sentence. How big is your market?

Third sentence, how much traction do you have? Ideally this sentence is saying something on the order of, "We launched in January and we're growing 30 percent month over month. We have this number of sales. This amount of revenue. This number of users." Very simple. If you can't speak to traction because you're prelaunch, you need to convince the investor that you're moving extremely quickly. "The team started working in January. By March we launched a Beta. By April we launched our product." Convince the investor that you guys are moving fast and that this isn't some long slog. You guys aren't thinking about this like a big corporation. You're thinking about it like a startup where you can move fast and make mistakes. That’s all you have to do in 30 seconds. Three sentences. From that basis you should be able to start a conversation about your company. From that basis I understand exactly what you do. You have no idea how valuable it is to be able to explain to someone what you do in 30 seconds. Internalize that. If you take nothing else away, that's going to help you.

Ok. Two minute pitch. Now you got someone you actually have to convince of something. Maybe even someone you have to ask for money. So I like to add four additional components. And these also go by very quick. The first is unique insight. Now if you talk to VC's they'll say stuff like, "What's your secret sauce? What's your competitive advantage? What's unique insight?" It's all the same thing. When I think about unique insight, what I think about is here's your opportunity to tell me something that I don't know. Here's your opportunity to tell me something that the biggest players in the market you're trying to enter don't understand. Or don't do well. This is the AHA moment and you better have it down in two sentences. The AHA moment. So you got to crystalize all the reasons why you guys are going kill the competitors or the really intelligent thought that got this business started in two sentences. And I need to AHA. You can see whether it's happening when you're saying it. That's why I like two sentences so you get in and out fast. So if I look at you and I'm like, "Uh." Then it's ok. You nailed it. If I look at you and I'm like, "I already knew that." Then you didn't nail it. If I looked at you and I just don't understand what you're talking about you definitely didn't nail it. So practice that unique insight. In your two minute pitch that's all you’re going to get - you're only going to get two sentences to get that out there. So it can't be complicated. And that's basically the theme of this whole thing right? It cannot be complicated.

Next - how do you make money? You know your business model. I see so many founders run away from this question because they think things like if I say advertising people are going to be like "Oh that's stupid." Just say it! Don't run away. If it's advertising - say advertising. Facebook's a massive advertising business. So is Google. If it's direct sales - it's direct sales. If it's you know a game and you're selling in app add ups - like that's fine. Just say it. Don't run away from the sentence. It only has to be one sentence long. Where founders get tricked on how you will make money is they say, "Well - we're going to run advertising. Maybe some virtual goods. We're going to figure out how to this. And maybe this. And maybe this." Well now you're saying nothing. Now you've told me you have no idea how you monetize this. This was a check mark that I just wanted to write. And then I am going to monetize it - instead I am writing a bug question mark. So do the thing that everyone else your industry does to monetize 95 percent of the time - say it and move on. Like it's totally ok. No one’s going to hold your feet to the fire and say three years later you didn't monetize this way. But it's much better to be clear and concise than it is to start spouting out every single way your company can make money.

Then next one is team. I think that this answer is actually really clear. I think you're trying to do two things. If your team has done something particularly impressive - you need to call that out. "We were the founders of PayPal." Probably want to say that. "We were the founders of Amazon." Probably want to say that. So if you guys have done something that is made investors money. You want to say that. If not, then please don't go on about the awards your team has one or the PhDs - I don't care. I don't care. What we want to hear is how many founders. Hopefully between two and four. We want to here is how many them are technical? How many engineers versus business people. Hopefully it's fifty/ fifty of more engineers. We want to hear is that how long have you guys known each other? We don't want to hear that you guys met a founders dating an even three days ago. Ideally you've known each other either personally or professional for at least six months. We want hear is that you're all working full time. It's really helpful. We're all committed to this business. And what we wanna hear is how you met. That's it. You can get in and out of that two sentences very easy. Your only way to build credentials is if you have accomplished something. And with an investor, typically if you accomplished something that's made someone some money. So don't try to over inflate yourself if you don't have that stat on your resume. Move on. The more you talk about a bad thing - the worse it looks.

So the last one is the big ask. When it comes to this, you have to figure out whether this is a conversation involves fundraising or not. What I tell people is like this is the time where you kind have to know what you're talking about. This is a time where you have to know are you raising on convertible note. Are you raising on a safe. You have to know what the cap of that safe is. You have to know how much money you're raising. You have to know what the minimum check size is. These are things where if you don't know these things, investors going be like, "These guys aren't serious. Or they haven’t done their homework." So where's the rest of this whole thing you shouldn't use any jargon. This part you shouldn't just be like "Oh we're just raising some money." Now is time to actually use a little bit of that jargon. If you don't know that jargon - Google search it. Like it's real simple. You'll guys learn it fast. That's it. That's all your pitch. Done. Game over. Now you let them talk.

When to fundraise? This is important. You've got this little growth graph here. Investors like to invest based on traction. It is literally always better to raise money when you have more traction than less. Often times though, you will be in a situation where you're just starting or you just launched. What you need to do is you need to think about how you flip the equation. Your entire mindset should be: you are the ones asking investors for money and therefore they are strong and you are weak. How do you create a scenario where you are strong and they are weak? That's where you want to be fundraising. First, how do you know that you're strong? If investors are asking to give you money, you're strong. That might be a good time to start fundraising. If investors aren't asking about giving you money, are you talking to people about your start up? Or are you running super stealth? If you're talking to people about your start up and you're getting the word out, either through the press or just through talking to your friends or people you know doing startups, that's a good way to start feeding that.

The second this is, have you created a plan so that you can launch and grow without needing to raise a bunch of money? 95 percent of the startups that I meet can get a product to market with a very little bit of money. Never put the investor in the ultimate position of power. "We can't do anything until you give us money." You always want to flip it around. You always want it to be, "This thing's moving. We all left our jobs. We're all working full time and it's moving. If you want to jump on, great. If not, there are a lot of angel investors." That's the attitude you want to have. That's the confidence you want to have. If you need money early, always plan on needing less money. Always be able to show that you've got a fully committed team that's working fast. That's going to be how you gain an advantage when you can't show traction. If you can show an investor that you haven’t launched yet but you've done eight months of work in one month or two months and you've got a great team that have all quit their jobs and they're totally committed, then you get some advantage back. You don't get all of the advantage unless you have launched and are growing.

Finally how to set up investor meetings. This is really, really simple but I'm surprised at how many companies don't get this right. The first is you want a warm introduction from another entrepreneur preferably. Or a previous investor of yours.

That's where you want to start. If someone who's past on your company as an investor offers you to make introductions that's kryptonite. Don't touch that. So first warm introduction. Very simple. You don't want to cold call these people. You don't want to bum rush these people. The person - the credibility of the person who is introducing you to an investor is big part on whether the investor will take that meeting.

Second, think in parallel. So many people that I meet will run the fundraising the super slow process. We met with one guy this we. We're going to schedule a meeting with another guy next week. Another guy three weeks from now. When you're fundraising you're on. It's a sprint. It's not a marathon. So you want to schedule all of your meetings during the same week. It's extremely hard to do but here's one trick that I love - tell when you're emailing investors you getting those warm intros the investors email you back you say, "Hey we would love to set up a meeting but we're building like crazy for the next two weeks. So can we set it in that third week?" Right? So then you've emailed everyone that. Right? So everyone schedules that meeting three weeks out. It's better for them because their calendars open. It's better for you because you've got all you meetings in one week. And also what did you do? You hinted, "Hey. I am not desperate for the money. We're building. Like I can meet you in three weeks but we're building. We're busy." Like it's signally all of the right things. So, that's the best way to kind of go about how you're gonna do that. The last thing is one team member should be investing in fundraising full time. It shouldn't be something that takes over the whole company. Because it's very, very distracting.

So with that - let's kick it off to the next part of this. Who am I handing it to?

Dalton: Hi. My name is Dalton Caldwell. I'm one of the partners at YC and one of the things that we're going to do today real quick is a mock pitch. And first of all I know this is a bit contrived. This is - in this format of like a college class, we're going to do our best to have fun and kind of demonstrate what it's like. And I realize there's a million reasons why this - why you can say, "Of this isn't realistic of what pitches really like." But again there's a lot that we can show you.

Just in terms of my background - over my career I've raised 85 million over several companies so I've sat in a lot of investors meetings. So I'm going to be pulling as many things as I can. So again, we're just going to try to show you something to talk to and use it as a learning session. You already did your intro earlier Qasar right?

Qasar: I've done a couple of startups.

Dalton: Cool. We're going to do two pitches and go through them pretty fast. As Michael said, these tend to go fast. Let's go dive into the first one.

Qasar, I understand you're coming to pitch me today. What can you tell me about what you do?

Qasar: We're building a communication platform that will allow businesses and consumers to collaborate on one single platform rather than in the fractured state that they're in right now.

Dalton: I don't follow.

Qasar: Think about WhatsApp or Snapchat. Those are for consumers. We want to do that for businesses. I have to do this with a straight face. What that means is we want to enable consumers to talk to businesses. That's the goal of our business or what our startup is.

Dalton: Who uses this product? What does the product do?

Qasar: It’s for consumers and businesses. A messaging product that allows consumers to send-

Dalton: Why would a consumer want to use your product?

Qasar: Because they want to message a business.

Dalton: What can you tell me about the market and the opportunity? What's the size of this company?

Qasar: Messaging companies are really big obviously. WhatsApp sold for 19 billion dollars. Snapchat is really growing very quickly as well. We think the opportunity is very big.

Dalton: Can you tell me a little bit about your traction, your numbers. Have you given this to people yet?

Qasar: We don't want to open the kimono and go into all the details here. I had a high level hour live, we definitely have thousands of users in the Bay area. Hundreds of businesses.

Dalton: Can you tell me who some of those businesses are?

Qasar: There's ones that you've been to. We don't really want to get too much into the details because we're still early, we're trying to stay stealth.

Dalton: Ok well, can you tell me about what you've learned so far. What insights that you've had from the customers...

Qasar: Yeah the consumers are sending messages to these businesses. And we think that's great. So and these businesses are responding to the messages and we think that's - I don't think that's obvious that would happen.

Dalton: So can you tell me about what your business model is and how...

Qasar: Yeah so we, we charge businesses like a monthly rate. We haven't precisely figured out what that is. We - right now we're free for the few hundred companies we're in right now. But we’re looking to probably do a monthly...

Dalton: How much do you think a business would be willing to pay?

Qasar: We thing certainly ten to fifteen thousand dollars a month...

Dalton: Ok. So anyway can you tell me a little bit about your team and who you have working on this.

Qasar: Yeah we have five founders. Technically I am the only one who's full time. Right now. We're raising money. So we can get you know the rest of the team on board. Yeah

Dalton: Can any of the founders program or...

Qasar: Yeah. I mean we have - one of them has a Bio PhD but he's really picked up coding. The - I am a python developer. I did - I learned python the hard way.

Dalton: Look at the time. Well it's been really great meeting you. Please keep me in the loop. This sounds fantastic.

Qasar: I will send you an update.

Dalton: Just keep me in the loop as this progresses.

Qasar: I'll send you an update. Great. That was awful.

Dalton: Ok. So let's go through.

Sam: That's disturbing.

Dalton: That was obviously not strong. Let's talk about some of the mistakes. First of all, you need to make sure the person you're talking to knows what you do.

Qasar: This seems really simple but it's not.

Dalton: So many times people get flustered. They get nervous and they start talking really fast. There's no way you're ever going to convince anyone of anything if they don't know even what your app actually is. You have to know your numbers obviously. If you're very vague or evasive, don't even have a meeting. If you don't feel comfortable telling an investor what your numbers are, don't even meet with them. It means you're not ready yet.

For market size, try to give some plausible bottom up analysis and don't just name drop big companies that aren't even related to what you're doing. People tend to do that a lot. Try to have insights. Try to convince me that there is something that I don't already know about the market that I learned talking to you. Also, why are you working on this? Why are you suited for it? Is it a good thing to do? Finally, he didn't drive the conversation anywhere. Obviously that went poorly and he just let the conversation flail around until I cut the meeting because I ran out of time as fast as I could.

That was not a good pitch. Let's try that again.

Qasar. Ok. Let's do this.

Dalton: Qasar, I understand you have a company. Can you tell me a little bit about what you guys do?

Qasar: Yes, we're a messaging product. That's kind of vague. What we allow you to do is essentially message a location. When you walk into a Crate and Barrel, you can send the Crate and Barrel manager a message like, "Hey. There's puke in the hallway." Or if you're in the airport "I am trying to find this specific gate 'cause I am not at this airport, "Where is the terminal for Virgin?" Or if you're at Target, "What aisle is the shampoo in?"

Dalton: So is this a mobile app?

Qasar: On the consumer side we have an iOS and Android app but getting consumers to download apps is obviously very difficult.

Dalton: I don't usually download app just to send a message to Crate and Barrel.

Qasar: Most businesses have a call to action which says text the owner directly. We tested a bunch of copy that works the best in small print. In small print we have the messages are anonymous. They also lower the barrier to entry. I think that most counter intuitive then we've learned in the kind of launch that we've had - in three hundred fifty locations in Bay. We've been doing this for about three months. We're about 11 percent weekly growth rate in terms of requiring businesses but most counterintuitive thing that we learned - Because we weren't actually sure is - Will people send messages while they walk it work...

Dalton: Do people send messages

Qasar: and they do.

Dalton: Like what's the number one type of message that people send?

Qasar: So originally we started the product thinking this is going to be like in location feedback. That was the premise. In location feedback. What we found is more than half the messages are actually not about feedback at all. They ask things like, "We were in this location in San Jose - this khaabob stand - Father and Son and we say messages that went through the satellite like are you hiring? And that's like very strange because you would think like why wouldn't you just ask the owner? But we realized that we know this is the owner and the person who's walking in doesn't and so they do prefer to actually just text the owner because I think that's an easier reading.

Dalton: Ok so it's like a suggestion box. It's like a way to just like message a business

Qasar: Initially that's what we thought what it was. But what we actually discovered was vast majority of - I shouldn't say vast majority. Over half the messages are just things like, "When do you open? When do you close? 'Cause that's not on Google. Do you - are you catering? Do you have any reservations available tonight?" etc.

Dalton: Ok look - in terms of your traction is sounds like you said some businesses. Like tell me about what you guys have right now.

Qasar: So we have three hundred and fifty businesses - all from San Jose to San Francisco. We sold them ourselves as three founders. We're all technical but we actually did all the sales because we learned a lot about how these businesses work. We actually come from a retail background. We originally built this product for large enterprise players like Starbucks and Walmart but we recognized at closing those contracts and our limited amount of runway wouldn't really be possible. So we wanted to get the product in the hands of users so we did S&B's. And that's when discovered, hey this like messaging product...

Dalton: Ok that sounds interesting. It sounds like you have customers. How can this be big though? Like ok - maybe you can get whole thousands of words....

Qasar: So in terms of like numbers - we see one and half messages on average per location per day. That might not sound a lot but for a business that's getting thirty messages - you take like a Yelp review or a Google review in a life time of business they might get five or seven. So they're getting a huge volume of messages relative to what they tend to experience and they're private so they are not public. So in terms of how do we actually make money, it's not - you know frankly speaking we don't have a very clear answer there. The two pats are the S&B side or the LC side the large customer side. Large customers we know from a retail experience just regular feedback tools are are three to four million per per year. So like a Sears - where we came from. S&B's we've tested are willing to pay 50 dollars a month. So I, you know certainly I think this is - can be a large business but there's clear ways to make money but...

Dalton: I can see that. Just a couple things. Like, can you tell me about distribution strategy and also just a little bit about the team

Qasar: Yeah, so distribution - so the thing that we learned in selling through these S&B's is really freaking hard. The formula LTV minus CPA - Life time value minus Cost Proposition A in S&B is never going to work out. So we have two solutions - one is to go up market like we originally planned to Starbucks or Walmart's. Or two is actually essentially pair with consumer facing companies Yelp, Google, Facebook...

Dalton: Have you been talking with them. Are they going to actually do it?

Qasar: Yeah - so we've talked to Google and Facebook. We're meeting with the Apple. We're basically want to introduce every time you search for a business there should be a message button. We want to get consumers in the habit of knowing they can send essentially a text message to any business. That can help us get broad distribution. Our real vision is to become kind of that infrastructure - that messaging infrastructure between consumers and businesses. If that doesn't work - Let's say Google, Facebook and Yelp don't want to give up that valuable property - it's really an add unit. We do just want to sell this an s feedback tool to large players.

Dalton: Alright. Can you tell me a little about the team - we're running low on time.

Qasar: There's three of us. All technical. Mike and I did a company before. Sonny was an ex school engineer. We come from retail. So our first start up was a failure. So I don't know if that's good or bad. We've worked together - we're all technical. We all built everything ourselves. And we sold everything ourselves.

Dalton: Ok.

Qasar: So we already had a couple of conversations with your firm. We're raising five hundred thousand on an 8.5 million convertible note. Of that five hundred two hundred -fifty is committed by Mike Maples, Eli Gill and Aden Sinket. And Mike with Floodgate is willing to fill the round. We think you're - you particularly - you and your firm can bring a lot to the team with your retail experience. Is this something that's interesting to you?

Dalton: Yeah - you know I think this is really interesting. I mean I would need to talk to a couple of more folks on my side but I do think that this - this could be pretty big.

Qasar: Yeah since we've had a couple of conversations before and we're certainly willing to meet again. We are closing a round this Friday and so certainly take time and let you other partners know. I will be available between now and Friday. I'll give you another call before Friday before we close the round. But we've love to actually see you - see you in the run.

Dalton: Ok. Well it sounds good. I got to go but thanks for that

Qasar: Great. Thanks.

Dalton: So in terms of that one you know - some key points here is try to actually tell a narrative that makes sense to people. You noticed there was narratives there talking about people - how they really use it. We were able to like tie it down to the real world. Which is good. He was able to demonstrate insights and actually tell me something I didn't already know about the market. Like there were some tid bits. It was more of a collaborative meeting where it felt more like a conversation than just like I was interviewing about something in my opinion. He actually asked for money. You saw I could have easily been just like, "Ok. Got to go." But he did talk about fundraising as Michael mentioned. And he was able to provide all he context and all the the questions to actually have a serious conversation with him. If he was KG about it or shy about and clear on the numbers there's a very good chance I probably would have just ended the conversation due to time pressure.

Qasar: Yeah. It's interesting we sit on this side a lot. You really - you can tell when people are very passionate and know their business very, very well. And that's what you have to become.

Ok so closing thoughts here before we - what you want to do after the meeting. Before we get into Q&A. We're running a little short on time.

After the meeting the first just like Tyler said in the sales things follow up. This is important. Anything other than a check or wired funds is a no. So they we got to keep talking to partners - I assume that's a no. And so you do want to put some pressure. The way you can do that is get deal heat. A deal heat is just a term that means there's a demand to be in your round. This is the easiest way and important way to drive a price, etc. Do diligence on investors, So let's say you have that five hundred thousand to raise for your seed round on the 8.5 million like we used as an example, Do diligence on the investors - If you do find - I do the diligence on Dalton and I found that hey he's actually not great investor, I can get Millan or Mike Maples or whoever to actually fill the rest of the round. It's uprising to us how money entrepreneurs don't do this. You would - it's like you would actually spend a lot of time hiring somebody - you're selling a part of your company to somebody you should know who you're selling it to to make sure they're the type of people you think they are. And then last - know when to stop. So some founders get so good a fundraising they just want to it all the time because it's much easier to do than actually building the company.

Dalton: Fundraising does not equal success. Nobody realizes that. We'll say this now but I am sure that everyone will still equate fundraising with success and read about someone's fundraising and assume that means they're successful.

Qasar: My intuition about why this is true is because a lot of smart people applied to good schools and to good jobs and they think fundraising is just another application that they can check off. Building a company is much more ambiguous.

Sam: Can you guys just stick around for a few minutes after to answer questions?

Thank you guys very much that was great!

Lecture 20: Closing Thoughts and Later-Stage Advice

Sam Altman

Good afternoon. Welcome to the last class of How to Start a Start Up. This is a little bit different than other classes, which have been about things that you should think about at the beginning of a startup. Today we're going to talk about things that you don't have to think about for a while. Since I'm going to not going to get to talk to most of you again before you get to post-product market fit stage, I want to give you the list of things that you need to think about as your startup scales. The list of the things that founders usually fail to make the transition on.

These are the topics we're going to talk about. Again, these are not writing code or talking to users, which means with a few exceptions that I'll try to note, you can ignore them until after you have product market fit. For most companies, these things become important between months 12 and 24. Write these down somewhere and look back them when you get there.

The first area we're going to talk about is management. In the beginning of a company, there is no management. This actually works really well. Before 20 and 25 employees, most companies are structured with everyone reporting to founder. It's totally flat. That's really good. That's what you want because at that stage, it's the optimal structure for productivity.

What tricks people is when lack of structure fails, it fails all at once. What works totally fine at 20 employees is disastrous at 30. You want to be aware that this transition will happen. You don't actually need to make the structure complicated. In fact, you shouldn't. All you need is for every employee to know who their manager is and for everyone to have exactly one manager. Every manager should know their direct reports.

You ideally want to cluster people in teams that make sense but the most important thing is that there is a clear reporting structure and that everyone knows what it is. Clarity and simplicity are the most important things here. Failing to do this correctly is really bad. Because it works in the early days to have no structure at all, it feels cool to have no structure. Many companies are like, "We're going to try this crazy new management theory and have no structure." You want to innovate on your product and your business model.

Management structure is not where I would recommend trying to innovate. Don't make the mistake of having nothing, but don't make the other mistake of having something super complicated. A lot of people fall into this trap. They think people feel cool if they're someone’s manager and if they're just an employee, they don't feel cool. So people come up with convoluted circular matrices management structures where you report to this person for this thing, and this person for that thing, and this person for that thing, while this person reports to you for this thing. That's a mistake.

This is the first instance of an important shift in the founder’s job. Before product market fit, your number job is to build a great product. As the company grows past 25 employees, your main job shifts from building a great product to building a great company and it stays there for the rest of your time. This is probably the biggest shift in being a founder.

There are four failure cases we see all the time as founders become managers. So I am going to talk about the four most common ones. The first one is: "being afraid to hire senior people." In the early days of a startup, hiring senior people is usually mistake. You just want people that get stuff done, and the willingness to work hard and aptitude matters more than experience. As the company starts to scale, and at about this time when you have to put in place the basic management structure - it is actually valuable to have senior people on the team. Executives that have built companies before. Almost all founders after the first time they hire a really great executive, and that executive takes over big pieces of the business and just makes them happen - the founder says, "Wow! I wish I had done that earlier!" But everybody makes this mistake and waits to long to do this. So don't be afraid to hire senior executives.

The second mistake is "Hero Mode". I will use the example of saying someone that runs the customer service team.

Someone who runs the customer service team -- they want to lead by example. This starts from a good place. It's the extreme of leading by example.

It's saying, “You know what? I want my team to work really hard rather than tell them to work hard I'm going to set an example. I'm going to work 18 hours a day. I'm going to show people how to get a lot of tickets done." But then company starts growing. They have the normal discomfort of assigning a lot of work to other people. So the company starts growing and the ticket volume keeps going up. Now they're have to do like 19 hours a day, and then 20 hours a day. It’s just obviously not working. But they won't stop and hire people because they're like, "If I stop even for one day we're going to get behind on tickets." The only way to get out of hero mode in this case is to say, "You know what? We're going to get behind on tickets for two or three weeks 'cause I am going to go off and I am going to hire three more support team members. I've calculated based off our growth rate that this is going to last this long. Next time I'm not going to make the same mistake. I'll get ahead of it and hire again."

But you actually have to make a trade off. You actually have to say, "You know what? I need to hire more people and we're going get behind on other stuff." That is the right answer. The wrong answer is to stay in hero mode until you burn out. Which is what most people do.

Third mistake: "Bad Delegation". Most founders have not managed people before and certainly have not managed managers. The bad way you delegate is you say, "Hey, employee, we need to do this big thing. You go off and research it. Come back to me with all the data and the tradeoffs. I'll make a decision and tell it to you and then you go off and implement." That's how most founders delegate. That does not make people feel good and it certainly doesn't scale.

A subtle difference but really important is to say, “Hey - you're really smart. That's why I hired you. You go off. Here the things to think about. Here's what I think. But you make this decision. I totally trust you. And let me know what you decide." That's how delegation actually works. Steve Jobs was able to get away with the former, and make every decision himself and people just put up with it. Every founder thinks they're the next Steve Jobs. A lot of people try this. For 99.9 percent of people, this second method here works a lot better.

Then the fourth area -- it's just a personal organization one. When you are working on product, you don't actually need to be that organized in terms of how you run the company and how you talk to people about what they're working on. But if you fail to get your own personal organization system right - where you can keep track in some way of what you need to and what everybody else is doing and what you need to follow up with them on - that will come back to bite you. Developing this early as the company begins to scale is really important.

Two other things that we hear again and again from our founders they wish that they had done early: simply writing down how you do things and why you things. These two things - the how and the why - are really important. In the early days, you just tell everyone. "Employee, when you're sitting around having lunch or dinner, you know this is how we think about building product. This is how we push to production. You know, this is how we handle customer supper."

Whatever. As you get bigger you can't keep doing that. If you don't do it, someone else is just going to say it. But if you write it down and put it up on a Wiki or whatever that every employee reads, you as the founder get to basically write the law. And if you write this down it will become law in the company. And if you make everyone read this - as the company hires a hundred and then a thousand employees - people will read this and say, "Alright. That's how we do things."

If you don't do it, it will all be random oral transition of whatever the hiring manager or their best friend that they make it their first week in at the company tells them. So writing down how you do things and the why -- the why is the cultural values. Brian Chest talked about this really well. Every founder I know wishes they written down both of these - the how and the why- earlier to just establish it as the company grows. And then this becomes what happens. It's one of the highest leverage things you can do that people don't.

Next area - "HR". HR is another thing that most people correctly ignore in the first phase of start up because, again, it's not writing code. It's not talking to users. But it’s a huge mistake they continue to ignore it. The reason I think most founders ignore it is they have in their mind this idea of like TV sitcom HR, you know. Awfulness. But it doesn't have to slow you down. Actually it speeds you up.

Most founders will say out of one side of their mouth, "People are our most important asset." And the other side, "We don't want any HR." So what they mean is that we don't HR - we don't want the bad kind TV HR. What good HR means is a few things. A clear structure. Which we already talked about you know a path for people about how they can evolve their careers. Most important, one of the most important things is "Performance Feedback." Again, this happens organically early on. People know how they're doing. As the company gets to 25, 30, 45 people - that gets lost and it doesn't have to be complex. It can be super simple. But there should be a way that it happens and it should be frequent. People need hear pretty quickly how they're doing. It should tell if they are doing badly to where you get them out of the company. Or if they're doing well it should. There should be a clear path to how this ties to compensation. That's the next thing.

In the early days of a startup, people compensation is whatever they negotiate with the founder and it's all over the place. As you grow - it feels hopelessly corporate but it really is worth putting in place these "Compensation Bands". So a mid-level engineer is in this range. A senior engineer is this range. Here's how you move from this to this. It keeps things really fair. Someday everyone will find out everyone else's comp. If it's all over the place, it will be complete meltdown disaster. If you put these bands in place early you will at least be fail. It will also save you a lot of crazy negotiation.

One thing that I think is really important when it comes to HR is equity. Most people get this right now for the early employees. They give a lot of equity. But you should continue to give a lot of equity all the way through. And this is one place that you investors will always give you bad advice. I think - not YC. But all other investors give bad advice here. Most do. You should be giving out a lot of equity to your employees. Now this dilutes everyone.

Right? This dilutes you as the founder and the investors equally. For some reason founder usually understand this as good. Investors are very short-sighted and don't want to dilute themselves so they'll like fight you over every equity grant. But, we've seen a lot of data at YC now and the most successful companies - and the ones where the investors do the best - end up given a lot of stock out to employees. Year after year... After year.

So I tell founders, "You should think about for the next ten years you're going to be given out 3- 5 percent of the company every year 'cause you just get bigger and bigger. So the individual grants gets smaller but in actuality it's a lot of stock. This is really important to do if you value your people you should be doing this. Specifically, you need to do this with refresher grants. And you should get a plan in place for this early. You never want an employee in a place where they vested 3 out of their four years in stock and they start thinking about leaving. So you should ALWAYS stay in front of peoples vesting schedules. And you know how they plan early where you have refresher grants in place.

There are a lot of new structures that people have been using here. I personally like six year big grants - but six years of vesting. 'Cause I think these companies take a while to build. There's pyramid vesting where you back weight someone's grant. In year four they get a lot more of the vesting than year one. There's a concept - different names for it, but something like continuous forward vesting where people's grants are automatically re-upped. Every year. At the same number of share. Whatever you decide, get an option management system in place at about this point. The normal way people do this is just someone keeps an Excel spreadsheet. I have seen mistakes that have cost employees or companies tens of millions of dollars because they didn't get this right.There's really good option management systems or software and you should get those in place around this point.

The other sort of HR stuff to touch on - there are a bunch of rules that change around 50 employees. Common examples are that you have to start "Sexual Harassment Training and Diversity Training". There's a bunch of others as well. But just put a little pen in your mind that when you cross 50 employees there's a new set of HR rules that you have to comply with.

"Monitoring your team for burnout." Again, it's up to product market fit. It's just a sprint. Now it becomes marathon. At this point you actually don't want people to work a 100 hours a week forever. You want them to go on vacation. You want them to have new challenges and do new things. And if you let the whole company get burned out all at once - that is often a company ending thing.

This is also a good time to put in place a "hiring process". Another thing that most founders regret is they don't hire - as soon as everything is working, you should hire a "full time recruiter". If you do this early - that's bad 'cause you'll hire too fast. That usually implodes. But most founders get behind the ball on this. There are a lot of sort of hiring process tips.

For example, I think most companies - even til they get up to say 3 or 4 hundred employees - should announce every offer on some internal mailing list or something before they make it. Because like half the time you do that. Someone in the company will know something good or bad about that employee. The companies that I know that have instituted this have been really happy. Also a good time to have a program in place to ramp up employees. So when someone starts, you know what their first week looks like. How did they get spun up? How do they learn everything they need to learn? Are they going to have a buddy that's going to think through them? That's going help them think through everything about the company.

Here’s one that you do need to think about before the 12 to 24 month mark. Which is "Diversity on the team." The most common place this comes up honestly is people that hire you know all guys on their engineering team for the first 15 or 20 people. And at that point you get a culture in place that sort of takes on a life of its own. Most founders that I've spoken to that have made this mistake regret it. They wish they had hired some diversity of perspective on the team earlier on. Engineering teams are not the only place where it comes up. But that's where you see it the most often, and if you get this right early, you’ll be able to grow the team much more quickly over the long term.

The other thing to think about is what happens to your early employees. So a common situation that happens is the company past the early employees. You know the company - you hire a engineer who's a really great engineer but then as the engineering team grows, you need a VP of engineering. The early engineer wants to be the VP of engineering. You can't do that, but you don't want the early employee to leave. They are an important part of the culture. They know a lot. People love them. So I think you want you be very proactive about this. You want to think about, "What's the path for my first 10 or 15 employees going to be as the company grows?" And then just talk to them about it. Very directly. Be up front, you know. Sit them down and say, "I want to see where you want to see your career go inside of this company."

Alright, so - "Company Productivity". This is something that you don't need to think in the early days because small teams are just naturally productive most of the time. But as you grow, it - the productivity - goes down with the square of the number of employees if you don't make an effort. Because it's sort of one these connections between nodes. Every pair of people add communication overhead. If you don't start thinking about the systems that you're going to put in place when the company is 25-50 people to stay productive as you grow - things will grind into a halt faster than you can imagine.

The second word that matters most to keep the company productive as it grows is "Alignment". The reason companies become unproductive is people are either not on the same page and you know don't know what the same priorities are. Or they actively working against each other. Which is obviously worse. But if you can keep the entire aligned in the same direction, you have won well over half of the battle. The way to start with this is just a very clear road map and goals. Everyone in the company should know what the road map for the next three or six months or a year - depending on where the company is in its life cycle.

You know a classic test that I love to give - is if I walk into a company getting - beginning to struggle with these scaling issues - I'll ask the founders, "Like, if I walked around and pulled 10 random employees and asked them what the top three goals for the company are right now - would they all say the same thing?" And 100 percent of the time the founder says, “Yes. Of course they would."

Then I'll go do it and 100 percent of the time, no two employees even say the same three top three goals in order. The founders can never believe it. Because they're like, "Well I announced it in all hands like three months what are goals were going to be. And how can they not remember?" But it's really important to keep reiterating the message about the road map and the goals. Almost no founder does this enough. And if you do it, you know the company will say, "You know, alright. These are our goals. We understand them and we're going to get them. “ Self-organize around that. But if people don't know what the road map of the goals are, it won't happen.

We already talked about figuring out your values early but I want to reiterate that. 'Cause that'll also really help company make the right decision. If everyone knows what the framework to decide it - they'll make hopefully the same decisions if they're smart people.

You want to continue to be run by great products and not process for its own sake. This is a fine, fine line. Because you do need to put some process in place. But you never want to put process in place that rewards the process. The focus has to always be on great product. One easy way to do this that a lot of companies try is they just say, "We're gonna ship something every day."

And if you do that - you know there's at least a continue focus on delivery. And then "transparency and rhythm" in how you communicate are really important. Most founders wait way too long on these but having a management meeting every week of just the people that report directly to the founder and the CEO - critical. All hands meeting - not quite sure how often is optimal for those. At least once a month. Where you go through the results and the road map of the entire company. Really important. Then doing a plan every quarter of what we're going to get done over the next three months and how that fits into our goals for the year - also becomes really important.

I put "Offsite" up there, because people don't do those nearly enough. A surprising number of the successful companies we've been involved with do a lot of off-sites. Where they take their best people for a weekend to a cabin in the woods or somewhere and just talk about what we want to be when we grow up. What are most important things to be doing? What are we not doing that we should be doing? But get people out of the office and out of the day today. Everyone I know that does thinks they're well worth the time.

So the goal in all of this productivity planning is that you're trying to build a company that creates a lot of value over a long period of time. And the long period of time is what's important here. You can avoid all of this and with the authority of the founder - make sure the company ships a great next version. But that won't work for version 10. It won't work for version 11. The single hardest thing in business is building a company that does repeatable innovation and just has this ongoing culture of excellence as it grows. If you look at the examples of this - most companies fail here. Most companies do one great thing where the founder just pushes to get it done and then don't innovate that well on follow on products. It really takes founders that think about how I am going to do this second thing - this really hard thing to get something like an apple that can turn out great products for30 or 40 years. Or longer.

Alright - these are super tactical "Mechanics". This is definitely just to put on a list and remember these things for later. Alright - in the early days. People basically ignore all accounting and maybe if they're lucky have a shoebox full of receipts. They certainly don't have anything that looks like a financial report. This is is a good time to get it in place. You know when things are working say month 18 or whatever - you can do this with an outsource person. Just say, "You know what? We like to get our books in order. We want to start getting audits every year. We want to start a relationship with an accounting firm." Easy to do. Definitely worth it.

This is also a good time to collect your legal documents because it's easy to fix things now. If you actually assign someone to go through and collect every agreement that the company has ever signed, then when your landlord tries to screw you out of your lease and no one can find the lease... Which happens like half the time somehow. Someone will be able to find it. Also, you're almost certainly missing something. Some employees didn't sign their PIAA or whatever and you'll find it now - it’s easy to fix now. It's gets really hard to fix like in the middle of your next round of financing. So again this is time to bring like a little of the order to chaos.

"FF Stock" is a special class of stock for founders that founders can sell in a later round without messing up the common stock valuation. It used to be that most people set this up right when they started the company. Founders fund sort of popularizes which is why it's called FF stock. But it became a really bad signal. Right that were obsessed with their own personal equity when the company had nothing - turned out to fail most of the time. So investors learned if founders pushed on this in the seed round, it was a very, very bad sign. Most founders don't actually want to sell stock until the company is worth like a billion dollars or something like. You can actually safely set this up after things start working in the next financing round and then you can sell it two, three, four years down the road. But it's a good thing to remember by around the time you get to the B round.

"IP, Trademarks and Patents". Actually just IP and trademarks. So, you have twelve months after you announce something if you want to patent it. And if you miss that window, it's very hard to do. So eleven months after launch or first publically talk about what you're doing - is a good time to file provisional patents. We recommend people just file provisional patents. All that does is hold your place in line at the patent office, and it gives you another year to decide if you want to patent something or not. It only costs about 1000 dollars. It takes way less effort than a full patent. And most of the time you'll know whether or not you'll need a fully patent a year later. But if you just do this one step, you'll at least have the option.

It's also a good time to file trademarks for the US and major international markets. Again, if you don't do this at this stage - most people end up regretting it. And while you're at it - a good time grab all the domains.

FP&A -- good time, also to think about someone to start doing FP&A. Most companies don't end realizing where they knobs on their financial model are until far too late. It turns out that if you have someone build a really great model of the business - and by really great, apparently Roelf Botha - who was the PayPal CFO and built their FP&A model - the top, like the top sheet of his spreadsheet was 15 hundred lines just a level of the detail people build these to. But you can really optimize the business and understand it at a level that most people totally miss. Most people don't hire someone like this until their many hundreds of employees. It's worth hiring earlier.

Another thing that I think is worth hiring earlier that almost no one does is a full time fundraiser. Let's say you hire someone really, really great and their full time job is to raise money for the company. You hire them after your B round. And you say, "You know what? By the time we raise our C round, we want the valuation be double what it would have been otherwise." You almost certainly get better results than if you hire an investment banker or someone else if it's just someone internal with the company. And you end up paying way less money and take literally half the dilution. This is one these slightly non-obvious optimization that people just fail to make.

"Tax structuring". This is another thing. Once things are working it would be worth you spending a little bit of time thinking about how you set up the tax structure for the company. I confess I don't know a lot about the details here 'cause I just find it personally really boring. But like if you assign the IP to some corporation in Ireland that licensing it back to the US Corporation. You end up paying no tax. No corporate tax. But I know that you can only do that relatively early on. And this ends up being a huge issue for companies that don't do it that compete with companies that do it you know once they're big public companies. So that's worth doing.

A lot of people through the class have talked about "Your own Psychology" as a founder. Here's what they haven't said. It gets worse. Not better. As the company grows you continue to osculate. The highs are better but the lows keep getting worse. And you really want to think about this early on and just be aware that this is going to happen. And try to, try to manage your own psychology through the expanding swing that it's going through.

Another thing that happens as you begin to be successful as you go from being someone that most people rooted for - kind of the underdog. To someone that a lot of people hating on. You see this first in internet commenters who will be like, "I can't believe this shitty company raised money. It fucking sucks. It's like awful. And it only bothers you a little bit. But then journalists that you kind of care about it start writing this and it just goes on and on. This also will go on and on as you get more and more successful. You just have to make peace with this early. But if you don't it will bother you all the way through.

This is also a good time to start thing about how long of a journey this is going to be. Very few founders think long term. Most founders think kind of a year in advance and they think, "You know what? In three years I am going to sell my company and either I am going to become a VC or sit on the beach or something." Because so few people make an actual long term commitment to what they're building - the ones that do have a huge advantage. They're in a a very rare flight class. So this is a good time to sit around with your co-founders and decide, "You know what - we're going to work on this for a very long time and we're gonna build a strategy that assumes that we're going to be doing this for the next ten years." Just thinking that way alone, it's probably a very high leverage thing you can do for success.

Take vacation. Another common thing that we see is founders will run their business for three or four years without ever taking more of a day of vacation. And that works for a year or two years or something like that. It really leads to a nasty burn out.

Losing focus is another way that founders get off track. This is a symptom of burnout. When you get really burned out on running business you want to do easier things or sort of more gratifying things. You want to go to conferences and have people tell you how great you are. You know what to do all these things that are not actually building a business. And the most common post YC failure case for the companies we fund is that they are incredible focused during YC on their company - and then after, they start doing a lot of other things. They advise companies. They go to conferences. Whatever. Focus is what made you successful in the first place. There are a lot of reasons people lose focus. But fight against that really, really hard.

This is a special case of focus. As you start to do well - you will start to get a bunch of potential acquires sniffing around. And it's very gratifying. You're like, "Wow! I can be so rich." And I'll be so cool. And MNA negotiations feel really fun. This is one of the biggest killers of companies. Is that they entertain acquisition conversations. You distract yourself. You get demoralized if it doesn't happen. If an offer does come in - it's really low. You've already mentally thought that you're done and so you take the offer. As a general rule don't start any acquisition conversation unless you're willing to sell for a pretty low number. Don't ever just check it hoping that you're going to have the one miracle high offer. If that's going to happen you'll know because they'll just make you a big offer before you can meet them. But this is big company killer.

And then - just a reminder to everybody - that things that kills startups at some level is the founders giving up. So sometimes you should quit but if you mismanage your own psychology and you quit when you shouldn't, that is what kills companies. That is the final cause of death for most of these startups. And so if you can manage your own Psychology in a way that you don't quit - don't get to a place where you need to quit or give up on the startup. You'll be in a far far better place.

So "Marketing & PR" is something that we tell companies to ignore for a long time. Everyone thinks in the early days that the press is going to be what saves them. We tell them all the time it doesn't work that way. It's definitely true. Press is not what's going to save your start up. But as you start to be successful - this is something that the founders themselves need to spend time on. So once your product is working - switch from not caring about this to caring about it a little bit. The two most important things for the founder to do - the founders to do - figure out the key messaging yourselves. Never outsource to your head of marketing or PR firm. You founders have to figure out what the message of the company is going to bet. And once you've set that it kind of sticks. Very hard to change this once the press decides how they're going to talk about you.

The other thing is getting to know key journalists yourself. PR firms will always try to prevent you from doing this because they need to have a reason to to exist... And so they're like, "We're going to handle the relationship with the journalist. We'll just bring you in for interviews." No journalist wants to talk to a PR flag ever. They're so much happier to hear from to just hear from the founder. The biggest PR hack you can do is to not hire a PR firm. Just pick three or four journalists that you develop really close relationships with that like you. That understand you - which you get. Then you contact them yourself; they will cover every story you ever give them. And they'll actually pay attention and get to know you and care about the company. This is so much better than the normal strategy of having a PR firm blast 200 contacts that never read their emails with every piece of news. This is something that I think is important to start doing.

This is also the time in a company when business development starts to matter. And so in the early days you can basically ignore anything that would be like doing deals. Except maybe fundraising and sales. This is a time when they're important. And everything or many things that you do like even fundraising. It falls under the category of doing deals.

So there are - here's my one minute crash course on this. There are five points that are important to understand here. We've talked about this a lot. Nothing will matter if you don't "Build a great product". So assume that you've done this before you go try to get anyone to do anything with you.

"Developing a personal connection" with anyone you're trying to do any sort of big deal with is really important. For whatever reason, most founders fail to this. Or many founders fail to do this. But no one wants to feel like they're this transactional thing. That you're using them to get distribution for your product or to raise money or whatever. So figuring out some way to actually care about this person and care about what you're doing with them. And not view them - you brave to in your own mind not just view them as this one off transaction. You have to actually care about them and what they're going to get out of this.

"Competitive dynamics" - this is a basic principal of negotiation. Most founders learn this the first time in fundraising. But it actually matters for everything. The way you get deals done and the ways you get good terms is to have a competitive situation. You don't do this deal with party A, you're going to do it with party B. It's not always an option but it usually is.

And this is the single thing that makes deals happen and makes deals move.

Tyler talked about "Persistence" -- the last lecture. So I won't hit on that again too much other than to say you go beyond your comfort point here most of the time as a founder.

And then the fifth point is that "You have to ask for what you want". This is another thing - I still have trouble with this and certainly most of the founders we do have. If you want something in a deal - just ask for it. Most of the time, you won't get laughed out of the room and might get it. But you have to be - at some point, you actually have to say, "You know this is what I'd like to do." Even if it feels aggressive or an over-reach or whatever.

So I am going to close this part of the talk with an image. One of the Airbnb founders drew this on like a business card or something for another founder that starting a company and then I saw it once and took a picture of it. 'Cause I thought it was such a good summary. And what he had tried to draw here was the YCombinator process as he remembers it. I love it 'cause it’s like so simple and it looks so doable when it’s written on a business card. But you're trying to find product market fit. You're trying to build a product and you're trying to close the gap between those two gears. The only way to do that is to go off and meet the people. You can't do this without getting really, really close to your users. And then he drew this graph that sort of on a white board that at YC and gotten kind of sort one of the YC rites of passage. But that's the graph of how adoption goes for a new company.

So you launch on the press. You get a huge spike. It falls off to nothing. At some point at least one point things look like they're going to completely die and kind of dip below the X access. They recover a little bit, you have this long, long troth of sorrow before things work. In Arabian B's case, it was a thousand days before the graph started taking upward. You have these wiggles of false hope. And then finally, finally, finally, finally things start to grow. Three years later. So starting a startup ends being this very long process. It is - it can be very rewarding. It's definitely long but it is doable. That's what I love about that drawing.

So with that. I have about ten minutes left. I can questions on this or anything else in the course that we've covered. If anyone has some.

Yes.

Audience member #1: You hold that diversity is important, but an earlier speaker said that diversity wasn't important and that you should just hire people that are very much like you and trust you...

Sam: So the question is how you square the device of diversity being important with earlier speakers saying that you want people that are very similar.

The difference is that what you want is diversity of backgrounds. But you don't want diversity of vision. Like where companies get in trouble is when they have people that think very differently about what the company should be doing or don't work well together. You don't want that. You do want hire people that you know and that you trust and that you can work with, but if everyone on the team comes from exactly the same background you do end up developing somewhat of a monoculture. Which often causes problems down the road. Not always. Some companies have been successful with that.

So what we tell people is hire people that you know and that you've worked with before. But try to hire people that complementary and aligned towards the same goal. Not people that are exactly the same. 'Cause you just get a better skill set.

Audience member #2: So what are some examples of ways to make up productivity on a personal level? How do you do that on a personal level and also on an advance level?

Sam: How to keep track productivity systems. So, the one I use which I actually thinks works really well is I keep one piece of paper with my goals for sort of three to twelve month time frame. And I look at that every day. And then separately I keep one page for every day of my short term goals for that day. And so if I need to do something in like a week I just flip forward seven pages and I write down. And then I also keep a list of every person and what they're working on and what I need to tell them and what I need to talk to them about. What we talked about last time. So every time I sit down with someone I kind of the full state and a list of things for that person that works really well.

Audience member #3: So we talked a lot about the startups growing but most startups fail. Any advice for how to fail gracefully?

Sam: Yeah. Yeah. Great question. We should have covered that.

How to fail gracefully. So, most startups fail and Silicon Valley almost goes too far on how it loves failure. Failure still sucks. You should still try not to fail. And this whole like thing of like "Ahh failure is great!" I don't agree with, but it will happen to most people most of the time and it's a very forgiving environment. As long as you are up front about it and ethical and don't let anyone get into bad situation. So if you're failing, first of all you should tell your investors, and second of all, you should not totally run out of money. What you don't want is blow up which a bunch of depths that the company owe and everyone showing up to work one day and the door being locked.

You'll know when you're failing and you'll know the company - things just aren't going to work. And you should just tell you investors, "Like hey. Sorry. This isn't going to work." No one will be surprised. Like I expect to lose my - or I'm willing to lose my money on every investment I ever I make. I know that happens most of the time and the winners pay for it you know still with a factor of a hundred. And so it's ok, No one - people will be very understanding and supportive. But you want to tell people early. You don't want to surprise them. And you want - you don't want to like let your employees get shocked when they know they don't have job. You want shut the company down in a graceful way. Help them find jobs. Make sure you give the two or four weeks of severance payment so they're not suffering a cash flow problem. All that stuff is pretty important.

Audience member #4: How many immigrant founders have you seen in YCombinator?

Sam: How many immigrant founders have we seen in YCombinator? In the last batch - I think it probably went up for this next batch. In our last batch 41 percent of the founders we founded we're born outside the US. From thirty different countries. So it's a pretty big percentage.

Audience member #4: I was just thinking - what do you think are the good places to start start ups?

Sam: Apart from the Valley where do I think are other good places to start a startup. Well I still think the Valley is the best by a very significant margin. But I think it's finally maybe beginning to weaken a little bit because the costs have just gotten out of control. To be clear - if I was going to start a company I still wouldn't think about it. I still will pick Silicon Valley. And think if you look at the data of companies of over the last few years that is to wins by a lot. But Seattle, LA - Lots of places outside the US - I think all of these makes sense.

Audience member #4: Like places outside the US?

Sam: I hesitate to make recommendations because I haven't spent enough time in the cities to really have an intuitive feel. But like - you know as well as I do the common ones people talk about start up hubs. I just can't make a personal recommendation there.

Audience member #5: So when should the founders start to thinking about hire a professional CEO - a senior guy?

Sam: When should the founders think about hiring a professional CEO? Never. You - if you look at the most successful companies in tech they are run by their founders for a very long time. Sometimes forever. Sometimes they even hire professional CEO and realize that is not going to like build a great company and so Larry Page came to be CEO again. I think if you don't want to be the long term CEO of a company - you probably shouldn't start one. I am not totally sure about that. I think there are exceptions. But generally that the transition that I talked about today if you go from build ing a great product to building a great company being a founder for nine of the ten years is going to be about building that great company and if you're not excited about doing that - I think you should think hard about it.

Audience member #6: What are some of the most common and alarming warning signs you should be looking for when you're trying to make the shift from building great product to building a great company?

Sam: What are the most common mistakes to make when you're shifting to building a great company? I think I went through most of them here. I tried to put everything here that I see people mess up most of the time. Yes.

Audience member #7: Is there a way to get involved in the Yom community before getting accepted?

Sam: Is there a way to get involved with YC before getting funded? No and intestinally not. I say the one thing you can do is if you work at a YC company and then later apply - I think probably like - well not probably that definitely if you get a good recommendation from those founders will help with YC. So you know, working at a YC company helps but there's not much you can do to help. And that's intention. Like there is no pre start up in a way that there is premed. You should just focus on whatever doing and then when you start a start up - there are `things like YC and others that are structured to help you. Most of the founders we fund we don't know at all before we do it. You know you really don't need to get to knopw us or get involved. We're all good that way.

Audience member #8: The statistic you saying now harder to get into YCombinator than getting into Harvard. So I am curious the criteria’s that you use to pick up startups. Does it change over time?

Sam: The question is what criteria to pick startups and has it gotten harder? Has it changed? The two things that we need to see are good founders and a good idea. And without both of those we won't fund the company. But that hasn't changed. That is always been the case. The applicant pool to YC has grown quite a bit. But most of - a lot of the growth is people who shouldn't be starting start up anyone that are just do9ing it 'cause it is sort of the cool thing now. So you know if you're really passionate about an idea and the idea is good and you are smart and you get things done and your we executing - I still think you have a very reasonable shot at YC even though the headline number is bigger.

Audience member #9: There's a certain market that you’re really excited about that don’t necessarily know all about yet - is there a certain track you recommend or ways to?

Sam: Sure - if there's a market that you’re excited about but don’t a lot about yet what should you do? Two schools of thought on this. One is to just jump right in. Learn it as you go. That's worked a lot of times. The other is go work at another company in the space or do something in the market for a year or two years. I lean slightly towards the second but as long as you are willing to really learn and really study and to get uncomfortably close to your users - either case would work. And I don't even thinks that it’s that much of a disadvantage. I think all things being equal go spend a couple of years learning about it in detail but I don't think you have to.

Audience member #10: I have a question related to YC - So I think YC did a fantastic job in promoting partnership in Silicon Valley. In fact, I plan to invest in some in the next three years. You guys pump 180 companies per year coming to the market it looks like its hard to follow each of the YC company any more. Do you think that this will create some - some people will walk away from YC because they cannot follow large batch of companies and the company had to be very polished and the firm had to be think of the world about ideas?

Sam: Alright so I think the question is do I think investors are going to fund less YC companies as we grow. No. Definitely not. Like certainly the trend in this is the other way. We have more and more investors saying that half their portfolio is not YC companies and they look forward to the day where it's three quarters. No I don't think that’s a problem at all. I think that so not on my top hundred problem list. The opposite of that maybe.

Alright. One more question.

Audience member #11: When should a group of founders raise a seed round or Series A?

Sam: In general it's nice to wait until you have the idea figured out and initial signs of promise before you raise money. Razing money puts some pressure on the company. Sometime pressure. And once you've raised money you can't be in this exploratory phase in definitely. You end up having to rush and so like if you haven’t raised money and your idea is not working you can fall around and pivot until you really hit on the thing that’s working. But if you've raised money and your `idea doesn’t work - You're in this oh shit moment. And you have to pivot and you pivot to whatever vaguely plausible idea is. And that’s bad. So I think if you can wait to raise any outside capital more than say like a hundred or 200 thousand dollars even necessary - but ideally not even that. Until things are working or at least pointed in the direction of working you're way better off.

Alright thank you all very much! This was fun!