GUIDE TO YC

The complete and indispensable game plan for founders applying to Y Combinator

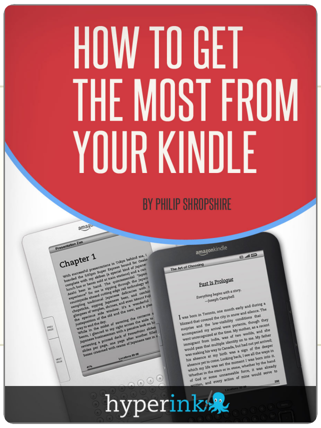
Jason Shen

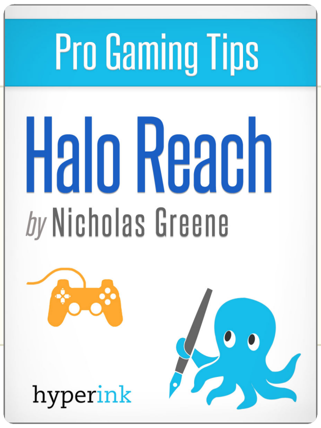
CO-FOUNDER, RIDEJOY (YC S11)

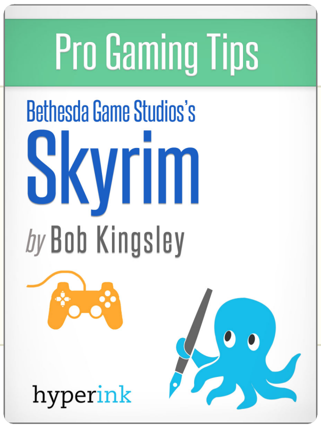
hyperink

Table of Contents

Other Awesome Books







Hyperink Benefits

- Interesting Insights
- The Best Commentary
- Shocking Trivia

- The Best Book on Designing iPhone and iPad Apps
- Pro Gaming Tips: Halo Reach
- The Best Book on Pro Gaming Tips: Angry Birds
- The Best Book on Finding & Developing Your Android App Idea
- How To Get The Most From Your Kindle Device
- Pro Gaming Tips: Skyrim
- The Best Book on Marketing Your Android App
- The Best Book on Kung Fu Tech Interviews
- The Space Elevator Project

View the full selection of Books at our store →

About The Author

Jason Shen

Jason Shen is the co-founder of Ridejoy - a Y Combinator-funded community marketplace for ridesharing. His blog, The Art of Ass-Kicking, has been featured by Lifehacker and ReadWriteWeb. Jason studied biology and philosophy at Stanford where he led the men's gymnastics team to an NCAA national championship.

Ridejoy is a company where friendly and talented people can do their best work and make the world a better place. We would love to have you join us: http://ridejoy.com/jobs

*Startup School cover photo c/o Robert Scoble (*http://www.flickr.com/photos/scobleizer/*). Follow him* @Scobleizer*!*

Get in touch:

I.

Introduction: Guide To Y Combinator

Acknowledgements

I’d like to thank **Paul Graham** for writing such wonderful essays, which have taught me to yearn for the vast and exciting world of startups (apologies to Antoine de Saint-Exupery).

My cofounders **Kalvin Wang** and **Randy Pang**, who have taught me strategic thinking, taste, humility and ambition.

The **partners** (and **staff**) at Y Combinator, the amazing YC **alumni** I’ve be lucky enough to know and learn from, the patient and supportive **advisers** and **mentors** who have helped us over the years, and the numerous folks who asked me so many great questions about their applications.

Readme

How to Read this Guide

This guide began as a series of personal notes to help friends and acquaintances interested in applying to Y Combinator. Its main purpose is to illuminate the YC process, and to equip you with ideas and approaches to bring out the best in your application. Read it as you would peruse extended notes from a long coffee meeting – and then go use it to make your application stronger.

Version History

v 1.0 (September 2011)

- Original notes drafted up in Google Doc
- Major chapters covered:
    - Why Apply,
    - Team,
    - Idea,
    - Customers,
    - Video,
    - Writing Tips, and
    - Recommendations

v 2.0 (February 2012)

- Transferred Google Doc on to guidetoyc.com, cleaned up for web viewing
- Fleshed out several sections, reviewed new sources (eg: mixergy interview)
- Added Resources section:
    - YC founders advice,
    - GiftRocket interview advice,
    - Links & Additional Reading

v 2.3 (March 2012)

- Worked with Hyperink to make digital version of the guide
- Revised various sections based on feedback from YC Partners
- Added a section for International Applicants, made various copy edits

PG Versus Other YC Partners

Paul Graham (or “PG”) has built a vast body of work around startups, hacking and business success, (among other topics) and so I often quote his writing in this guide. As one of the founders and partners at YC, I believe his views are influential in determining what kind of team, idea and market YC is interested in funding.

However, he does not have the final word when it comes to applicants.

There are ten partners at YC that read applications and conduct interviews; it’s already the case that not every partner interviews with every startup – meaning you may not even present in front of PG if you get a YC interview. Regardless of whether you sit across the table from two or twelve YC partners on interview day, if you’re wise, you’ll read up on everyone. It’s a solid interview strategy to figure out in advance how your application will impress the YC partners based on their experiences and individual world views.

Disclaimer

This guide is based on solely my personal experience and judgement, conversations with YC alumni, and things that Paul Graham and other people have written and/or stated about getting into YC. Act on this advice at your own discretion – if things go well, congratulations! If things go poorly, you agree to hold me harmless (though if you’re really distraught, I *might* buy you a beer).

The content in this guide (outside of the direct quotes) is not validated or endorsed by Y Combinator, Ridejoy or anyone else besides myself, Jason Shen.

Feedback

I have poured hundreds of hours over many months into making this guide as helpful as possible. If you have any questions, comments or feedback, I would love to hear them. I may not be able to reply to all of you, but I will definitely read everything. You can reach me at jason@guidetoyc.com.

II.

Chapter 1: Background

YC, Ridejoy, And Me

My name is Jason Shen, and I applied with two cofounders to YC's Summer 2011 batch – we were fortunate to get in. This was my first time applying to YC, and Ridejoy is my first tech startup. I'm nontechnical, but happened to have two technical friends (and roommates) who both worked at startups. I'm aware of how lucky I am to be in this position.

We got a lot of help from friends and YC alumni when preparing our application, so I wanted to write this guide to give back. Working with YC has definitely been a game-changer for me, and my startup would not exist with out it. I sincerely hope that this guide leads you to a similar place, and provides you the benefits I've received.

But the goal of this guide is NOT to help you "cheat" your way into Y Combinator.

The objective here is primarily to help people understand what I think YC is looking for and how to best convey the qualities that make for strong startups. I think the process of applying to YC in of itself can make your startup better, and that the partners are ultimately so good, there's no way you are going to "sneak in."

But, as Paul Graham says in How to Apply:

Like all investors, we want to believe. So help us believe. If there's something about you that stands out, or some special insight you have into the problem you plan to work on, make sure we see it.

My hope is that by reading this guide, you can avoid any common mistakes and pitfalls, and give yourselves the best chance of success. You can better help YC believe.

III.

Chapter 2: Why Apply

The Process

To Sum It Up: There are a lot of great reasons to explain why you should apply to Y Combinator – if you're reading this guide, you probably already have a few lined up. You might also consider YC for the following:

- Instant relationships to amazing people,
- Strategic, "insider" advice and perspective on entrepreneurship,
- A solid infusion of seed capital, and
- An ever increasingly recognized "startup stamp of approval"

As Y Combinator itself puts it:

The overall goal of YC is to help startups really take off. They arrive at YC at all different stages. Some haven't even started working yet, and others have been launched for a year or more. But whatever stage a startup is at when they arrive, our goal is to help them to be in dramatically better shape 3 months later.

What Happens at YC

Having matriculated through the program, I'll offer my take on the value to be found. It basically boils down to the following four things (in no particular order): **knowledge, network, resources and brand**.

1. Knowledge

The YC partners have an immense amount of startup expertise. For every single problem you face, or will face, as a founder, there have been numerous YC companies who have dealt with the same issues – and guess who helped them get through it? Office hours with the partners at YC aren't really like what you see on Techcrunch Disrupt – they're much better. And this is invaluable for your startup.

YC also has amazing speakers – on the caliber of Startup School, but more candid and with the chance to participate in a Q&A afterward. There are also special panels devoted to fundraising, and a lot of great prep before Demo Day. You'll have the chance to learn a ton of useful knowledge about startups – and specifically how your startup can be successful – by doing YC.

2. Network

When you get into YC, you instantly have 150+ (depending on the size of the batch) smart, successful, well-connected people who will, without a doubt, take and respond to your emails. This is in addition to the 800+ founders in the broader YC alumni network, who are both extremely helpful and generous.

This network is invaluable to the startup process: it doesn't matter if you are trying to hit an industry contact, figure out a nasty technical issue, decide between different service or software providers, backchannel to an investor, or just need someone who reads German – someone in your batch or in the full network can help.

YC alumni help each other to a greater extent than most any other organization alumni I have seen. It's a special thing and an absurd, almost unfair advantage for founders.

3. Resources

Y Combinator invests 11k + 3k per founder in exchange for 2-10% (but almost always 7%) and, while it's expensive as far as money goes, it's a bargain considering what you receive in return. In addition to the YC seed funding, since Winter 2010, every team has an offer to accept convertible debt from the following:

- Start Fund: $50k, led by Yuri Milner, a Russian billionaire
- SV Angel: $50k, led by Ron Conway, a well-known "super angel"
- Andreessen Horowitz: $50k, led by the super ex-entrepreneurs Marc Andreessen and Ben Horowitz

This note is on very entrepreneur-friendly terms, and almost everyone takes it. Getting into YC means that you, or your startup, now has at least a year of burn for your founding team, which is an amazing opportunity. There are also a number of additional perks and deals offered to Y Combinator companies.

4. Brand

Everyone in the startup world knows what YC is these days, and it's almost universally regarded positively. The only semi-negative is that some investors think YC companies are "overpriced," but this is only bad in the sense that you'll be unable to receive funding from investors with such opinions because your company’s valuation is too high for them.

In general, investors are much more willing to talk to you when they know you are a YC company. They may not invest, but they will almost certainly meet you (especially after Demo Day). More and more people outside the world of startups are hearing about YC, which is only a good thing for YC and its affiliates.

IV.

Chapter 3: Basic Advice

The Easy Stuff

To Sum It Up: Engineer your application (and interview) to optimize for rapidly impressing the YC Partners.

YC is pretty open about its application process – which is great. Paul Graham has written multiple essays about what they look for and how one should apply to YC. So the first thing to do is read this very, very carefully: How to Apply to Y Combinator.

Back? Great. Let's start on my first and most basic piece of advice:

Get to the Point Fast

Y Combinator received over 2,000 applications for YC in S11 and a similar number for W12. There are a total of ten partners who read the essays, and YC also asks some founders for feedback. But the sheer number of applications, paired against time constraints, means the first run-through of your app is very very short.

Tedious, hand-wavey and mostly illustrative math follows:

According to PG, every partner reads every app. Partners read about 700-1000 apps (after the first culling by YC alums) over a period of about 10 days. Assuming an 8 hour day with 5 minute breaks each hour means each application, on average, gets about 2 minutes of each partner's time. (440mins per day / ~85 apps per day = 5.2 mins per app)

The takeaway here is that you need to get to the point fast and make every answer matter. At the end of the day, nothing matters except that they say "oh shit, this team is really smart/over-the-top hardworking/resourceful/can get things done/likes each other/could successfully build a great startup." Think constantly about how you can impress with substance, not style (unless the style is in your product). Give numbers if they seem large, and explain things in the most favorable way possible.

Note: It's important that you don't ever lie or say something false. But, just as you don't discuss your work week in the same way to your boss as to your friend at the bar, be very mindful of how your application comes across. They need to first *remember* you, and secondly *think you're legit.* And you have very little time to make your case.

V.

Chapter 4: The Team

In Good Company

To Sum It Up: YC wants to fund great startups. Great startups come from great teams. Great teams are smart and technical, resourceful and tight-knit. They get stuff done.

Once I know what type of group I have, I try to figure out how good an instance of that type it is. The most important question for deciding that is:

Please tell us in one or two sentences about something impressive that each founder has built or achieved.

To me this is the most important question on the application. It's deliberately open-ended; there's no one type of answer we're looking for. It could be that you did really well in school, or that you wrote a highly-regarded piece of software, or that you paid your own way through college after leaving home at 16. It's not the type of achievement that matters so much as the magnitude. Succeeding in a startup is, in the most literal sense, extraordinary, so we're looking for people able to do extraordinary things. – PG (How To Apply*)*

PG says that they look for 5 qualities in founders: **Determination, Flexibility, Imagination, Naughtiness, and Friendship.** But what does this really mean? And how can you showcase these traits on your Y Combinator application? One more quote:

For most startups at this stage, the best predictor of success is the founders. So, the most important parts of the application for me are the questions about the founders' backgrounds and the most impressive things they've done. We're looking for evidence that the founders are smart, effective, and determined. – Sam Altman *(*How to Get Into Y Combinator*)*

The primary factor in determining the success of an early- (and even medium-) stage startup is the founding team. Your product can change, your market can change, but who the founders are isn't really going to change (unless someone leaves, which is usually bad, or at least disruptive). So YC really spends a lot of time evaluating the team when reading applications.

Evaluating The Founders

From going through YC, talking to partners and founders, and reading YC material, here's how I think evaluation truly breaks down:

1. Smart

Great teams are smart, so show YC your team is smart – meaning you are knowledgeable in the relevant fields of your industry, and about what it takes to run a startup. Show that you can learn new things quickly, can process information and make informed decisions. Sam Altman describes the application's set of questions about the business as "largely an intelligence test".

Obviously, YC considers traditional measures of intelligence (top-tier schools, brand name tech firms, winning Math Olympiad/Putnams, finishing college at 16, etc.) so don't be afraid to flaunt those things. But more importantly, they have demonstrated the desire and willingness to fund "unproven" folks – folks without pedigrees, that just have some other way of demonstrating how smart they are. This can be through past projects, deep product insights or other notable accomplishments; in our case, two of us went to Stanford, the other to Cal, and all three of us had worked at VC-backed startups.

Relevant questions on the app: Background, most impressive thing you've done, real world hacks, what do you understand about the market that others don't

2. Technical

Teams that are knowledgeable about how technology works are "technical" – you are presumably starting a company that leverages (and most likely produces) technology in some way. The assumption is that there is at least one person on your team who is pretty technical (i.e. can code/build most of the core product). But as time has gone on, more people without much in the way of technical backgrounds have been applying. I think it's much harder to get in without a strong technical background, but it's not impossible. Do whatever you can to emphasize how you're addressing this area – preferably by showing a barebones demo that you yourself built.

Don’t make it seem like you are waiting for a “technical cofounder” to save the day. It does appear that way if you’re building a “regular” web/mobile app, or something that doesn’t require crazy new tech; they probably won’t really dive into your technical chops if you have a cofounder with a CS degree or substantial experience building software or other technical products. But if you are shaky in this area, that is not a good sign.

On a personal note, I truly understand how difficult it is to do a startup without programming experience. I have done a fair amount of “cofounder dating” and have struggled through learning bits of Ruby on Rails without much success. I can only say that I was really, really lucky to find myself with two friends and roommates who were excellent programmers – and had the startup itch. It can happen, but it’s not easy. Then again, no one said forming a startup was a walk in the park.

Relevant questions on the app: Background, most impressive thing, hack, projects worked on together

3. Gets Stuff Done

Great startup teams have high output. They create things from nothing, and overcome hurdles to achieve their objectives. You need to show the YC partners that you guys will get stuff done, and ideally, that you have experience getting stuff done together in the past.

PG has said that a really bad sign for a startup would be to meet him in office hours, discuss a bunch of issues, questions and the like, and ten days later meet again to discuss the same things – as if basically nothing had really happened in between that time. This means the founders weren’t “getting stuff done,” and that’s very bad.

A personal story: my cofounder Kalvin and I had started and built a nonprofit together in college way before deciding to team up for Ridejoy, which was the basis for his “most impressive thing.” We also discussed the rather ridiculous startup roommate finding project that all three of us worked on, and how it provided us with experience brainstorming, planning, working and building together.

Relevant questions on app: Most impressive thing, hack, projects worked on together

4. Resourceful

This one is similar to being smart and get things done, but there’s also the element of something else – “clever” comes to mind, as does “naughty.” Willing to break the rules, find loopholes, tricks. You can’t always power through things, so YC founders want to see that you can find the back door, and that and you’re willing to maybe get in a little trouble to do something. That once you’ve found the back door, you’ll use it.

I’ve written a lot more about being relentlessly resourceful elsewhere on the blog.

Relevant quotes from PG:

“In any interesting domain, the difficulties will be novel. Which means you can’t simply plow through them, because you don’t know initially how hard they are; you don’t know whether you’re about to plow through a block of foam or granite. So you have to be resourceful. You have to keep trying new things.” – Relentlessly Resourceful

“Though the most successful founders are usually good people, they tend to have a piratical gleam in their eye. They’re not Goody Two-Shoes type good. Morally, they care about getting the big questions right, but not about observing proprieties. That’s why I’d use the word naughty rather than evil. They delight in breaking rules, but not rules that matter.” – What We Look for in Founders

5. Tight-Knit

People say that having cofounders is like being married with kids – minus the sex. You have this intimate connection in your baby (the startup) and you are pretty much willing to share your entire lives and devote all of your efforts to raising that kid/company.

YC really wants to see founding teams who have known each other for a long time, who have worked together and have a good fit. One thing that hurts startups is when cofounders bicker constantly, as it’s highly unproductive. Even worse is when they split entirely. (And this happens not infrequently, to both YC and non-YC companies.) This can tank the company completely.

PG lists “Fights Between Founders” as number 17 in Mistakes that Kill Startups, and says that 20% of the startups YC funds has a founder leave. That’s 1 in 5!

Do your best to show why you and your cofounders are going to make it. Why are you a match made in heaven? Will you stick together through thick and thin, even when the going gets rough, and you’re pissed at your cofounder, and you haven’t have enough sleep?

“Startups do to the relationship between the founders what a dog does to a sock: if it can

be pulled apart, it will be." – PG

Personal story: I lived with my cofounders for a year before we decided to do YC. We already knew each other's personality quirks. We had argued about stuff, dealt with money things, coordinated parties, and hunted for new roommates together. We knew what we were getting into, and we liked each other a lot. And we said all those things in our app. Now, not every team has the good fortune of so much history and synergy, but the more you can convey how tightly connected your team is and how your tendencies complement one another, the better.

Relevant questions on the app: How long have you know each other, projects worked on

VI.

Chapter 5: The Idea

Starting Up: Success & Impact

BREAKING NEWS: Y Combinator announced on March 13th that for the Summer 2012 applications, they would fund founders even if they didn't have an idea. This will likely be a controversial move, but as the article states:

"Our experience suggests that smart people who think they can't come up with a good startup idea are generally mistaken. Almost every smart person has a good idea in them. A good startup idea is simply a significant, fixable unmet need, and most smart people are at least unconsciously aware of several of those. They just don't know it. And we now have lots of practice helping founders see the startup ideas they already have."

This further reinforces the fact that demonstrating you have a killer founding team is a huge priority for YC applicants. Read: don't skimp on your idea, because a strong concept can really help your chances, but this definitely changes the game. A lot of smart, capable people will be coming out of the woodwork to apply to Y Combinator for the S12 cycle.

If you'd like more perspective on their experiment, you can review some of the comments on Hacker News.

To Sum It Up: YC wants to fund startups that will succeed and go on to play important roles in the Internet, technology, or society. You should also recognize that they have probably seen a bunch of teams apply with your exact idea. Act accordingly.

While the team is probably the part that YC thinks about most, obviously what you're working on is important as well. Here are some things to think about when writing about your startup/product is.

1. Success

"We want to fund founders that are going to be successful. If you can convince us you're going to be, you have a good chance." – Sam Altman (How to Get Into Y Combinator)

YC wants to fund startups that will succeed. This might sound obvious, but it's actually a really important point. However much you might want to do a startup or have an innovation you want to bring into the world, you have to remember that YC is NOT a charity. PG is not your rich uncle, here to help you pursue your dreams of being an entrepreneur. YC is in the business of funding winners.

In interviews, PG is very critical of the fact that most investors and venture funds fail to make returns. As an engineer/hacker, he finds it absolutely intolerable that they are wrong so frequently. Check out this exchange with Andrew from Mixergy:

Andrew: Really? Really!? So you still consider yourself bad at it?

Paul Graham: Oh god yes, we think of ourselves as just utterly terrible at picking these teams.

Andrew: Why? What's bad about it?

Paul Graham: Because our choices are always wrong! That's why. We have tons of evidence of how bad we are.

Andrew: What are your percentages of bad companies to good ones, would you say?

Paul Graham: At least a third are just disastrous, you know. In the venture business generally a lot of the investments are failures. Even a venture fund which has a lot more at stake and spends a lot more on effort in due diligence than our ten minute interviews. Even a venture fund half the investments will be failures.

So everyone in the venture business is bad, right? And maybe if we had more experience in the venture business we would take this badness for granted and think "oh well, actually we're really good if only half of our investments are failures", but we're not in the venture business so it seems intolerable. [emphasis mine]

– Paul Graham Mixergy Interview

I have previously speculated that YC is trying to figure out how to fund a high proportion of people who they can turn into winners. In other words, I believe they probably aren't thinking they can just find a few 10xers or 100xers and make up their returns.

If the founders themselves can be off by like many orders of magnitude about the exit, it's stupid to even think about it. You just want to fund people who are good and some of them will go public and some of them will just like explode on the starting line and there's *not much you can do about them. – PG, Mixergy interview*

PG's comments on this section:

I had previously written: "They truly would love to have every single company they fund be a "winner" even if there weren't as many mega companies."

This is the exact opposite of what we think. We're looking for outliers. That's why we like to fund crazy ideas. There's nothing we like better than funding an idea that has a small chance of succeeding, but will be huge if it does.

Of course, once we accept a startup, we want it to succeed. But that is a separate question.

I think PG sees startups as either win or lose in the long run, and if they win, the multiple (on how big that win is) is mostly up to chance. Since YC is investing at the same rate as the founders, for the same amount of time, it will be sold at as many X their investment.

So take a good hard look at your idea, product or startup.

Do you think this is going to work? No, I mean do you *really* think this is going to work? Do you have good reason to believe that you can REALLY make this awesome idea into a great company? You'd better or else spend some time doing something – prototyping, customer development, or exploring other ideas – until you are really sure. Because otherwise, how the hell are you going to convince the partners, who don't even know what's in your head?

2. High Impact

We'd be delighted to get an application that answered the question "What are you going to make?" with "A new search engine to compete with Google," so long as this was followed by "We know that sounds impossible, but we think we can get a toehold initially by…" Wouldn't you be interested at this point? Even if the plan had only a 1% chance of working, it would be worth backing. – Paul Graham *(*How to Apply to YC*)*

After optimizing for success, YC wants to fund companies that really make an impact on the world, that capitalize on the big opportunities that have emerged in society. They want Y Combinator to become an engine of progress for the world. You can see this in their Requests for Startups List – in the way they have identified problems that they want to "fix," like dating, news and "the problem" of Hollywood.

Here is an example of one of the interesting and BIG problems that YC is looking to solve.

Notice how they discuss markets, and the size and magnitude of the problem or opportunity. This should give you excellent clues as to how to write your own app:

RFS 5: Development on Handhelds

There seems a reasonable chance that handheld devices will displace laptops as development machines in the same way that laptops displaced desktops. What would it take? Maybe you could do it mostly with existing hardware. Maybe you'd have to make significant innovations in input and display devices.

The real test is whether you can create an acceptable development environment on something small enough that you'd be willing to use it as a phone. Whatever the solution turned out to be, the result would end up being useful to more than just developers.

RFS 5 – *Y Combinator*

Don't worry too much about hard numbers about the market, it just needs to feel relatively big. Most market sizing numbers are kind of BS – and they know it. Note that in none of their Requests for Startups or their older Ideas We'd Like to Fund posts do they include numbers.

Some of the best YC companies really are doing some amazing things – Airbnb is changing the way people travel, Dropbox is changing the way people collaborate, Weebly now powers something like 12% of the inernet, Heroku is a godsend for millions, Ruby (and now Node.js, and Java) helps developers everywhere who just want to deploy and not worry about server issues. If you can give YC the feeling that there might be a way for you to turn into that kind of company, you're set.

Starting Up: Concrete, Original, Desirable Ideas

3. Concrete

The fact that YC is interested in funding “big ideas” does not mean you should describe your idea as “fixing banking” (unless of course you are super baller, and you’re Simple). You’ve got to be concrete about what exactly you are doing now – and then make the case for why this could lead to something big.

Write as if you were explaining your company to a smart friend. Be specific about the problem you are trying to solve and how exactly your product will do it. It’s okay to talk in stages: “first we will build X to do Y, and over time we can turn that into a Z that does A, which is a bigger problem/market.” It’s also fine if your current thing does not sound super impressive, powerful, or fully-featured. YC understands that innovations often start off looking ugly, and even toy-like.

When you and the TechCrunch trolls make fun of startups launching their initial version, you’re like someone making fun of Dell when it was just Michael Dell assembling computers, or Microsoft when it was just a pair of undergrads writing a Basic interpreter. How could such companies possibly grow huge? – PG *(*HN Thread on Vid.ly*)*

Having said that, you still want to make your work sound as impressive as possible. If you have brand name clients, write them out! If you have really cool technology, flaunt it! If you have tons of signups, say so! And if you don’t: be honest about it, but explain why that’s alright for right now, and how you’re taking advantage of what you have done/sold/achieved so far.

4. Your Idea is Not Original

“‘X is terrible, we’re going to make a version that doesn’t suck’ doesn’t cut it but is surprisingly common. How specifically are you going to make it better? I’m looking for

clear and concise answers – that tends to be a sign of good founders.” – Sam Altman

There are only so many ideas. I mean, your idea has likely been tried by someone already – take a browse at springwise.com to see what I mean. PG and the YC partners have probably seen, heard, or thought about some version of your idea numerous times over the course of reviewing tens of thousands of applications and having countless discussions with founders and investors. The more general and consumer-y it is, the more times they’ve seen it.

Which means if you haven’t seen a YC company doing it, you’d better be wondering why. What makes your team, product, traction, or distribution strategy SO GOOD that they have to fund YOUR version of this idea? Better to say something insightful that demonstrates your unique ability to succeed. Do some research, talk to customers, have something concretely legit to show. This is really important, so take the time to think it through.

I used to be of the mindset that you should just “figure things out” as you go along and it’s not important to know everything about your problem space before you get started. Now, however, I believe that both for yourself and for impressing potential investors (like YC), you want to have a thesis when approaching your startup:

“Not only does X suck in these specific and important ways, but all previous attempts to solve X sucking have failed for these 3 reasons. Now, new technologies or social trends have created an opportunity for us to solve this problem using this new approach Y which we have experience and expertise in.”

5. Make Something People Want

The YC moto is “Make something people want.” This is so crucial that if your company is acquired, you get a special shirt that says “I made something people want.” PG believes that if you can do this, everything else falls into place and you can succeed.

7) Something your company needs that doesn’t exist. Many of the best startups happened when someone needed something in their work, found it didn’t exist, and quit to build it*. This is vaguer than most of the other recipes here, but it may be the most valuable. You’re working on something you know customers want, because you were the customer. And if it was something you needed at work, other people will too, and they’ll be willing to pay for it.*

So if you’re working for a big company and you want to strike out on your own, here’s a *recipe for an idea. Start this sentence: "We'd pay a lot if someone would just build a…" Whatever you say next is probably a good product idea.* – #7 from Ideas We'd Like to Fund

A lot of people think that YC is filled with consumer Internet-y startups. I think some of the earlier and more well-known YC companies started off the mold of Reddit, DailyBooth, Loopt and Posterous…all consumer-oriented startups. However, I believe that there are many more YC startups that serve business clients (or directly make money through product/service sales). These include Dropbox, Airbnb, Weebly, Stripe, HelloFax, MinoMonsters and others.

More on this in the next section, but basically, if you can show (ideally via some kind of traction) that you are building something that some group of people REALLY wants, then you are in a good place.

VII.

Chapter 6: Users & Distribution

Know Who's On The Other End

To Sum It Up: Deeply understand your users, make them extremely happy and know how you're going to find them. If you've got traction, show it off.

Users

Understand your users. That's the key. The essential task in a startup is to create wealth; the dimension of wealth you have most control over is how much you improve users' lives; and the hardest part of that is knowing what to make for them. Once you know what to make, it's mere effort to make it, and most decent hackers are capable of that. – Paul Graham *(*Startups in 13 Sentences)

Every startup has users and customers – how well do you understand yours? The problem that sometimes comes up for founders (especially those with a strong technical or engineering backgrounds) is that their product is a solution looking for a problem. Another is that they've built an overly-complicated product that doesn't really address the root needs of their users.

Show YC you understand your users. One question we, and many other startup, hear at office hours again and again is, "Who needs this? … No, who *really* needs this?" Your first answer is rarely good enough. You've got to convince the YC partners that you truly know who your users are, how to reach them, what their problems are, why this product fits into their life/workflow and how it solves their problems.

The thing we care most about in interviews (at least of things one can change) is how engaged the founders are with users. How do they know people actually want what they're building? Have they talked to real, live users? What have they learned from them? – PG, Questions You'll Probably Be Asked on Your YC Interview (HN Thread)

Let's look at the way YC tries to get at the answer to these questions (YC application):

Q (1/3): Why did you pick this idea to work on?

Be careful! This is not a question asking you to explain why you're personally passionate about X. Just because you love X does not make X a good idea. It's fine if you feel this way, but I would not make this your primary reason for doing this idea. It's better that you chose this idea because you know a lot about the domain and see a real, unmet need, either via personal experience or because you know certain businesses are willing and able to spend money to solve this problem.

Q (2/3): Do you have domain expertise in this area?

Ideally you have spent time studying this area, either academically, or through work experience/personal projects. It's not the end of the world if you aren't the known expert on this topic, but demonstrating any sort of expertise, even a highly-trafficked blog in this area, would be a plus.

Q (3/3): How do you know people need what you're making?

Ideally you can point to some interesting or surprising data or evidence to indicate that people really have this problem.

Distribution: How It Gets There

Q: How will you get users? If your idea is the type that faces a chicken-and-egg problem in the sense that it won't be attractive to users till it has a lot of users (e.g. a marketplace, a dating site, an ad network), how will you overcome that? – the YC application

Distribution is also extremely important. I'm guessing a lot of applicants haven't thought as much about this part of the app. That's ok – I think distribution comes after finding product market fit (the whole thing about users) but it's still a place where you can demonstrate you know your stuff.

Some of the standard distribution strategies:

- **Double-sided referral**: (á la Dropbox) Make sure this really applies to your business as it is everyone's favorite "free" marketing tactic, and doesn't always work.
- **Viral mechanics**: Some products give their users a very strong incentive (or involuntarily cause them) to share your product with others and can be a powerful way to gain distribution.
- **SEO**: In order to pull this off, you need to prove there is a lot of long tail search traffic (or huge volume in a few keywords) AND convince YC you can rank highly for the right keywords.
- **Advertising**: I would only bring this up if you can prove (with data) that you can acquire and monetize your users at a rate that is sustainable (Revenue per click > Cost per click) and scaleable, i.e. your targeted audience is sufficiently large.
- **Communities**: (blogs/forums/networks) Ideally, you are already an influencer in these communities, or you have the potential to get in there and quickly build strong relationships with the right people.
- **PR**: Some products are simply loved by the press (Hipmunk, Airbnb). Perhaps yours is too, but otherwise – unless you have amazing press contacts – this is likely not an easy option.
- **Direct sales**: This probably is going to be an important channel for enterprise level products or startups selling big-ticket items, like Carwoo.

- **Partnerships**: For this to work, there need to be strong partners with a high proficiency for distribution in your space and someone on your team who is a hustler, or willing to become skilled at business development.

If you DO have a chicken and egg problem (it sucks!), it'd be worth reading this great article by Ribbon Farm – Ubiquity Illusions and the Chicken and Egg Problem. It's long, nuanced and thought-provoking. A teaser:

If you want your metaphoric t's cross and i's dotted, the solution we are talking about is: fake the chicken while the egg incubates.

There are many ways to do this, most of them both stupid and illegal. For instance, you could doctor your resume and make up fake letters of recommendation.

A ubiquity illusion is a much more subtle mechanism, and in most cases, is not illegal.

Distribution is where you show you've done your homework. Marketplaces in particular have serious challenges when it comes to distribution. How do you solve the chicken and egg problem? The truth is that there is usually no answer *a priori*, you have to try a ton of shit and grind and hustle your way to user growth, but the answer you put on the app needs to show that you at least have some good ideas.

Here are two more useful links – but remember, there are no silver bullets.

- What is your customer acquisition strategy? (A guide to startup marketing tactics)
- Getting Users for Your New Startup

Traction

Remember, the unofficial official motto of YC is "make something people want," and traction is proof that you've built something users want. Users, beta clients, growth, partnerships, and revenue are all forms of traction. When I think about it, the motto is a condensed version of this:

Make something that many people want badly enough to build a viable startup around it.

Many people – this depends on the kind of startup you're building. If it's a free consumer web app, you really need high volume to show traction, or medium volume in a very specific niche (from which you could plausibly expand). If it's expensive enterprise software, you could do with much lower numbers.

Badly enough – having users is not the same as having engaged users. Do your users login more than once? Are they actually doing stuff on your site? Are they willing to pay for it? Do they share the product with their friends? Show YC how happy you are making your users. And remember what PG learned from Paul Bucheit: *It's better to make a few people really happy than to make a lot of people semi-happy.*

Viable startup around it – You need to convince YC that your startup's trajectory is toward success: that you are going to make more and more users happier and happier in a way that will enable you to build a legitimate business around it. Speed matters, and size of growth matters. Try to choose your strongest points of traction when writing your app.

VIII.

Chapter 7:Writing Tips

Filling Out Your Application

To Sum It Up: Make the writing in your application succinct, concrete, unique or memorable, and impressive.

And get lots of feedback! Writing a high-quality YC application is something between a college app, a scientific paper and a really good piece of direct mail. That is to say, you need to talk about yourself and your ideas in a clear and nuanced way, while striving to make the reader impressed and motivated to take action (in this case, invite you for an interview).

Let's look at the four major characteristics of the writing in a YC application:

Be Succinct

Write as concisely and clearly as you can, please don't use big dense blocks of text. – Harj Taggar, How Should I Fill Out the YC Application Form? (Quora)

This is a throwback to Chapter 1. YC partners are looking through hundreds of applications, and you don't want to waste a single word when they skim through yours. Try to keep sentences short, and avoid convoluted clauses and sentence structure. How would you explain it to a smart friend of yours if you had only ten minutes?

A great way to keep your writing short is to do a brain dump by writing everything you want to say, then going back and cutting mercilessly – multiple times. The word limits are a great guideline – it's ok to go over a little, but those extra few words had better be damn good.

Concrete

Don't use marketing speak e.g. "revolutionizing the world of digital communication via the social web" gets me no closer to understanding what you're actually doing. Write in plain speak e.g. "a mobile application showing you nearby restaurants your friends recommend." – Harj (again from the same Quora answer.)

Be matter-of-fact in your writing. Stick to data and solid points that can be backed up with evidence if needed. Include hard numbers when you can. This helps YC partners get a handle on what you're talking about. Ideally these are numbers generated through your product or through your own research – third party data is better than nothing, but much easier to question.

Concreteness matters because it increases the chances that the partners will "get it" right away. Remember, they will spend about five minutes (on average) reading your application. If YC rejects you, let it be because they understood what you were proposing and disagreed, rather than because they were just confused by your ramblings.

Unique & Memorable

This is a 2nd order objective. You should focus on making sure your app is concise and concrete. Having done that, it would help to make your application both unique and memorable. If you can cite surprising stats about your market, interesting stories about your resourcefulness, or something to that effect – do it.

You want YC to remember you because if you get an interview, you don't want to have to start over explaining the idea, who you are, or anything about yourself. Ideally, when you walk in they will say something like "Oh you're the ones who XXX." Try to make that XXX a good one.

Impressive

Realize the question asking about the most impressive things you've built or achieved is one of (possibly the) most important question on the form. Don't answer with "This startup", "I haven't achieved anything impressive yet" or use it as an opportunity to show your sense of humour. We're not looking for resume credentials here, examples of where you were determined to do something and stuck with it – regardless of the outcome – are impressive to us (since so much of what we're looking for in founders is determination). – Harj (one more time)

Always remember that your goal here is to get them impressed by your team, product and ideas. I've said this before, but I'll say it again: you want the YC partners to think: *"Damn, these guys are freaking smart, determined and onto something really interesting and big. We don't want to miss funding them."*

Oftentimes, hackers avoid thinking about impressing other people. This is a mistake. You've got to sell it. If there are facts that make you look good, even if they're only anecdotes or single pieces of data, throw them out there!

Other Random Thoughts

- The more traction you have and the more legit you are, the less the app matters. This means that if you are early-stage, it's worth putting more effort into your app. But if you are later-stage, it might be better just to try and get as much traffic as possible so you can submit great numbers.
- You can split up the writing at first, but make sure to go back through and ensure it has a unified voice.
- Don't skimp on the "interesting things you have learned" question. Additionally, if you have other startup ideas that seem good, do take a minute and put those in there. This is a chance to impress, or to be memorable in a positive way.
- Everyone should put in a "what you hacked on" answer, even though it only asks for one.

Finally: Get lots and lots of feedback on this app.

You've got mentors, right? Get as many trustworthy and intelligent eyeballs on your written application as you can. (This goes for any application for anything, really.) More eyeballs will always help you flesh out your concepts, spot weak links and strengthen your case.

Ask other people to read your application and ask them if they can honestly understand basic things, like what you're actually building. A lot of things that seem obvious to you, won't be obvious to someone reading your application for the first time. – Garry Tan, Super last-minute advice for startups applying for Y Combinator

If you get in, this will prove to be one of the most important pieces of writing you've ever done. So, swallow your pride and ask your friends to read it and leave detailed feedback. Mix it up between your techie friends, and ones who just happen to be good writers. Ask for thoughts on everything – the idea itself, how you frame things, the specific examples/facts you cite, and even grammar and punctuation. Revise, and send it to some of your friends again, and (if you have time) send it out for a second round to some new people, so you can get new first impressions.

IX.

Chapter 8: Demonstrations

Your Video

To Sum It Up: The video and the demo are similar in that they are both concrete, non-textual ways for you to convey who you are and what you've done. The same rules apply: try to impress YC as quickly as possible without falling into marketing-speak or overblown statements.

My advice for the video isn't terribly specific, so I'll just list some thoughts:

- They probably only watch the video if they like the rest of the app, since it sort of breaks the flow of the reading.
- PG said he added the video component to get a better sense of how the founders communicate and relate to each other.
- Basically, you'll want to introduce yourselves, give a one-sentence description of what you're making and then either describe:
    - Why you're doing this startup
    - What you've learned since starting
    - How you know each other
    - Expertise you've developed in this domain
    - Anything else that might make them want to fund you
- Ideally you're all in the frame, but it's ok if you record a Skype video – we used Call Recorder because Kalvin was traveling when we shot the video. I would avoid having just one founder in the video if at all possible.
- Don't worry about making the production value that high – as long as you can be heard and seen, I think that's fine.
- Having said that, human beings naturally tend to like attractive, happy individuals, so if you can find good lighting, wash your face and not wear your rattiest t-shirt during the

video, it can only help you.

We did a lot of takes of our video, but made sure not to memorize it so that it would still seem natural. You should have an idea of what you want to say. Ours had a lot of one person talking over the other person, some jokes and laughing, and we spoke pretty fast so we could squeeze in more information.

Check out this link to help you out a little further: HN Thread on Directed Edge Video.

Your Demo

My thoughts on the demo are that it doesn't actually play a major role in YC's decision. In general, they want to know you are a technical team that can build stuff – so it's good if you do have a working, live version of your product, and if you are getting it in front of users. But again, time is the big issue. It takes a while to click around a site and really use it enough to get a good perspective on it.

So unless your idea is super technical, it's unlikely that YC will spend too much time on the demo. If you are building a general web or mobile app, I think YC assumes that if you are technical, the issue won't be "can you make this?" So your mileage may vary.

PG's comments on this section are as follows:

The demo can matter a lot. It depends on the type of startup. If you're in a situation where you ought to have something to show for your work so far, it's bad not to. And no matter what your situation, it helps greatly if you have a great demo.

X.

Chapter 9: Recommendations

Make The Most Of Your Connections

To Sum It Up: YC is counting more and more on their network to scale their application process – leverage that via alumni recommendations.

People ask questions all the time about whether YC can handle the increased number of startups it is funding. The fact is that it's less about the ratio of partner to startup than it is the environment YC has created – and there is a decent system behind YC now.

But what I want to focus on here is recommendations: make sure you try your hardest to get some.

Reach out to YC alumni. They are pretty accessible – on HN, Twitter, personal blogs, etc, or get an introduction through a friend, if necessary. Write up a few sentence summary of what you're building (like a very compact version of the YC app) and send it along, asking for 15 minutes of advice on a few specific questions.

Don't just go into the call pitching – we'll see right through you. You need to be truly curious and open to feedback for this to work. (In general, when you ask people for feedback, you should request that it not be sugar-coated, and as critical as necessary. Sometimes people feel weird giving strangers strongly negative feedback, but this feedback can be exceptionally useful.) Ideally you should reach out as early as possible because those available to advise you will get swamped later on.

Be genuine and respectful. If they seem positive and supportive about your startup, then gently ask if they can write you a recommendation. Be nice about it – you're never going to bully someone into writing something nice about you. We had several YC alumni go up to bat for us and write us recs, anywhere from one or two sentences all the way to 5 paragraphs. In our interview, Jessica actually said "oh you guys are ____'s friend", where ___ was a YC alumni who had written us a really great rec.

PG’s comments on this section:

Actually it would be better to expend that effort on your company or the application. Don’t spend a lot of time spamming YC alumni trying to collect recommendations. They don’t carry any weight with us unless they are very strong. We can tell the difference between a recommendation that’s being made for our sake (because the alum thinks it would damaging for us to overlook the application), and one that’s being made for the sake of the applicant (e.g. because the alum is benevolent and they begged him to recommend them).

XI.

Chapter 10: Final Advice

Things You Should Do

rocamboleh: Is anyone aware of the number of submissions to YC per year and the chance of getting an interview?

pg: I realize this will sound like a flippant answer, but I mean it seriously: the chance is either very high or very low, depending on how good your application is. It took me a long time to realize that when the odds of getting into something were described as e.g. 1 in 10, that didn't mean the odds for any given applicant were 10%, but rather (to the extent the people deciding were good judges) that for 10% of applicants the chance was nearly 100%, and for the other 90% nearly zero. – HN Thread

Y Combinator is an incredible program and an insane asset for any entrepreneur. My startup would not have half (perhaps more!) of the success we've had if it were not for YC. Having said that, it's important not to get hung up on getting into YC. Remember:

You don't need anyone's permission to be an entrepreneur.

You do not NEED to get into YC to start a great/successful company. Just ask Mark Zuckerberg, Andrew Mason, Dennis Crowley, Aaron Patzer, Jaakko Iisalo (creator of Angry Birds) or scores of others. Look, I'm with you – when we applied, I really, *really* wanted to get in. And, yes, there's a magical quality to the Tuesday dinners, and the entire experience. But it's not a prerequisite for being an entrepreneur.

The good news is that doing the Y Combinator application is a great way to think through your business. It forces you to give real answers to important questions that will impact your business, which is useful – even if you don't get in YC. Force yourselves to give real (versus BS) answers to these questions. Read Paul Graham's articles on What Startups Are Really Like and Startups in 13 Sentences and see if what you're doing, and how you're approaching your startup and the app, are in line with these thoughts.

Make sure you are truly ready to commit to the startup life. Being a founder is a tough road. You will likely fail, and even if you do succeed, it will take far more time and effort than you anticipate. Yet it can also be one of the most fulfilling things in the world.

You must believe, to the core of your being, that you have the right team and you will make this happen no matter what anyone – including YC – says.

A Final Thought

PG has imaged YC as an institution for some time, and now has begun talking about it like a Harvard or Stanford. YC is something with legacy that's not just about startups, but about bringing together smart, ambitious, hardworking people who add value to the world. This is a natural extension of his thought that startups should Be Good.

If you are a capable person who's ready to put forth incredible effort and brainpower toward building a great company that makes a big impact on the world, one that creates value for customers and shareholders – then Y Combinator wants to fund you. I hope this guide will help you submit the strongest possible application and show Y Combinator why they can't afford to pass you up.

XII.

Chapter 11: In Their Words

Unorthodox Advice From YC Founders

I've asked some of my YC founder friends for their most surprising or interesting pieces of advice for getting into YC. Here's what they came up with:

Jeff Widman – PageLever (Facebook analytics)

Your interview is about selling, never forget that. Anytime you're selling, it's an emotional decision. But these are highly rational people, so you must back up the emotional sell with facts so their brain gives permission to their emotions.

We were a marketing agency who executed very well (helped YouTube become the fastest growing fan page in all of Facebook) before we decided to build a product company to solve things that frustrated us.

When we interviewed with YC, we had a very very basic product, and they turned us down because they were worried that we'd struggle to productize our consulting. They thought we were smart, they just weren't sure we could switch from clever-one-off campaigns to a product that helped all customers be clever.

We came back next interview round with a sheet of paper with big brand logos on it and said "These guys use our product." I remember they were all like "Wow that's impressive."

Once they quieted, I flipped it over and it had an even more impressive brands list of users. I distinctly remember PG saying "you've got YouTube? how did you ever get them as customers?" That magician's trick of double-sided paper is what did it for us. We also had the facts: 100 brands using it who had more than 1M Facebook fans, the % of sign-ups active on a weekly basis, etc.

Anand Kulkari – Mobile Works

1. All startups are broken. Don't feel bad if yours is broken. Be confident that you're going to fix those things.

2. Remember that the goal is NOT to get into YC. It's to build a successful company. If you are a successful company, you can be that without YC. Ask yourself before applying: what is your next goal as a company, and how will YC help you achieve it?

We needed funding to keep our company during the summer. It turned out we also needed solid mentorship and examples of what to do right. We got those from YC, but our company would still exist even if we didn't. We went in with the attitude that we wanted YC but that we would survive without it.

3. When we started out, we had ten users – and none paying. We'd done a long trial proving our model worked, and an enterprise customer who had agreed to a trial. But what we had more of was hustle and dedication and a killer, killer team.

Traction is proof that your model works. But team + commitment beats traction any day.

An Unannounced YC Founder

Network your way to past founders that have already made YC a lot of money. Get them to write you recs. I think these recs are much much stronger than just any YC founder.

Someone once told me that PG looks for three things in a founder. From most to least important:

- Persistence,
- Not being delusional,
- Intelligence.

I've thought about that a lot. I know so many people that have #1 in spades, and are at least adequately #3, but totally fail #2. At the end of the day, the partners are human, and they too have insecurities about their startup (Y Combinator) and their accomplishments as every founder does. Colin Powell said that you are an excellent leader if people will "follow you anywhere if only out of curiosity." That is the reaction you want to get out of the partners.

PG told me once that Jessica is YC's secret weapon, and that the decisions are largely made by going with her gut judgement of the character/personalities of the founders. I suspect that's an exaggeration (or possibly I misremember), but indeed, at the end of the day, your goal in the interview is simply to convince the partners, AS PEOPLE, that it would be fun to hang out with you for a few months.

I really don’t think that product or traction is that important, except insofar as it reflects on you as a person (e.g. if you have built a crappy product, not only does it mean you suck at building products, it also means you are either too clueless to realize it, or too lazy to care, or incapable of improving yourself, or delusional and unable to cope with ego-damaging information.) I would rather go in with nothing than with a crappy product. Whatever you have should be wicked impressive. Control reality – deny them access to any information about you that isn’t insanely great. I would rather go in with a product I built in 48 hours over the weekend immediately before my interview that is mediocre but roughly functional (which is what we did), than to go in with something that is “pretty good” but that took me six months to build.

A corollary: If YC accepts people, not companies, then they must also reject people, not companies. So if YC rejects you, focus on improving yourself. Become more persistent, less delusional, and smarter (or at least more focused, which is a pretty good substitute). That they accept people that they previously rejected, to me, is just more proof that people can improve themselves if they make the effort.

Treat your interview like you’re managing the media. Have a list of talking points in your head, and take every question they ask you, answer it, but then seize the initiative and use it to get out another of your talking points. (Just be sure your talking points are good and are genuinely interesting. Ask yourself: ifI have only a few minutes with these people, what’s the MOST IMPORTANT thing I could say that will make them not want to miss the chance to meet me “while I was still early enough in my career to care about YC?”) In particular, make sure you get to the demo with enough time to spare.

Scott Milken – MixRank

It’s hard to speak in generality, since our sample size is exactly one, but anecdotally some of the things we did right when applying with MixRank include:

- My co-founder had strong domain knowledge of the business we were getting into.
- I had domain knowledge with web crawling, which was very useful for building our ad crawlers.
- We picked a large market with an obvious and proven revenue path.
- We both went full-time several months before applying and bootstrapped product development, buying our own servers, etc.
- We started talking to customers even before writing code, so that by the time we applied we intimately understood the customers and the market.

- We stretched “MVP” to an extreme, and shared proof-of-concepts with potential customers as soon as possible.
- We pitched everyone that would listen (in particular, many YC alumni), encouraged critical feedback, and (eventually) addressed all major concerns.
- When we applied, our demo wasn’t nearly as important as our understanding of the business, all of the players involved, market size, threats, etc.
- We didn’t need the investment, and in fact, YC was an after-thought.

Some of the things we didn’t have when we applied:

- Customers, traction, or revenue,
- A live product,
- Any other investment,
- A long history together as co-founders (just about 4 months working together at that point),
- Our awesome domain, MixRank.com, or
- Any formal business structure; LLC, C-Corp, etc (which was a good thing– put this off as long as possible)

An Unannounced YC Founder

1. Practice the interview with someone who’s gone through YC. It’s surprising to us the number of people who don’t practice the interview because it’s not like any interview you’ve gone through. Having PG and team cut you off when it’s unexpected will totally throw you off.

2. Be able to clearly say what you’re doing in a sentence. YC reads a lot of apps, and if its not clear in a sentence what you do, then the idea is probably not clear in your head.

An Unannounced YC Founder

I was all about the non-technical hack question, even though I didn’t think it was important at all – it was just the only one that I had the competence to formulate an answer for – and I think our answer was hella funny.

I always used to ask Starbucks for a Venti cup, fill it up with nonfat milk (this is before I turned lactose intolerant) and chocolate powder (I also wasn’t anti-sugar yet), stir, and – BOOM! – “cold chocolate”, free drink! (Not on the Starbucks menu, FYI.) Now, I do the healthier version: I get a venti hot water from Starbucks, unscrew the top to the cinnamon, pour in a SHITLOAD of cinnamon, stir, and make bomb-ass cinnamon tea.

(Also free!)

My boy Yuri (not Milner, the YC homie) told us that's the FIRST PART of the applications that he reads and decides how carefully to read the rest of the application based on that answer.

Ilya Lichtenstein – MixRank

Application:

- Like everything with raising money, who you know and networking matters a great deal. When I came to Silicon Valley I didn't have a cofounder, I didn't know a single person in the Bay Area, and I didn't go to a top-tier school. By the time we were applying to YC, about 30% of the W11 batch was advising us on our application. So anyone else applying should have no excuses.

- People in the startup scene and YC alums in particular are really friendly. Set up an intro to them and they will help you. Cold email them – many will respond, and help you too. The secret to cold emailing: think only about how you can help them/add value to them. Studies show reciprocity is the single most powerful persuasion technique. We're genetically wired to reciprocate in kind – this is amplified through Silicon Valley culture. For example, I offered to help them with marketing ahead of Demo Day. When we were applying, many alums were happy to help.
- Brevity is incredibly important. Cut every answer in half, and then cut it again. The biggest mistake people make is trying to sell the big vision in the "what are you building question." Don't – just say simply what you're building.
- As you know, the team is more important than the idea, so spend a lot of time on questions about the founders and what they've accomplished. People spend too much time stressing about pitching the idea.
- Commitment and determination is massive. Quit anything else you're doing and focus 110% on your startup. Scott quit a high paying cushy job and I shut down an affiliate business making $1000/day to do this startup. Talk about this in your application.
- Traction trumps everything, even seemingly fatal issues, like not having a cofounder. So above all, keep working on your product.

Interview:

- 90% of success is showing up. Don't dither or stall, and have a quick, reasonable sounding answer to every question. The YC partners are likely not domain experts in your market, so anything plausible-sounding will work. Don't BS (unless you're very good at it), as they've learned to detect that, but don't freeze and rack your brain for specific facts either.

- They're looking for a mix of flexibility and determination. When they suggest a pivot or different idea, run with it – either explore the new idea with them or explain/justify why it's not as good an idea as it seems, preferably because you've already tried it and it didn't work. However, sometimes they will deliberately suggest bad ideas – stand your ground and don't be a sycophant to everything they say.
- If they like you, the interview will morph into the first office hours, brainstorming new ideas and next steps for the company. This is a very good sign. At this point, just go along with them and don't blow it.
- Appearances can be deceptive. I thought we bombed our interview completely, so right after we went home and started drinking heavily to drown our sorrows. When PG called, I was very drunk. This might not have been the best scenario, as our call went something like this:

PG: We'd like to fund you.

Me: Uhh...OK. Cool, dude.

Advice For International Applicants

An Announced YC Founder

If you're from one of the countries that apply for the Visa Waiver Program, you can get 90 days at a time in the U.S. without much trouble. This can help you get over here for any YC interviews with enough time to go through the YC dinners.

[The founding team] does not have full visas – we have used the visa waiver program about six times each, which means returning to our home country for at least a few days, then returning to the U.S. Of course, this is risky – but not illegal. Immigration still has the right to send you straight back home and give you a lifetime ban on using the visa waiver, but only the visa waiver; you can still apply for a full visa if you're banned.

Most foreign founders seem to either apply for the O-1 or the H-1B visas. O-1 works especially well if you have some sort of background in your field, as it doesn't necessarily require a degree.

Don't take this as straight fact, everyone's situation changes the rules they need to play by, so it's always best to do your own research and speak to an immigration lawyer. Most immigration lawyers should give you some basic advice for free, usually only charging you once you start actually applying for visas.

XIII.

Chapter 12: Supplemental Materials

Deconstructing A Successful YC Application: Proxino/Taazr

Originally published on Ethanfast.com on October 10th 2011.

The October 10th deadline for YCW12 approaches, and I expect that readers might appreciate a deconstruction of the successful application Muzzammil and I turned in last summer. Before I begin, a word of warning: At the time, we were both undergraduates, and we hadn't yet built our product. I'll go through our responses one by one.

What is your company going to make?

We described exactly what our product would do.

Taazr detects bugs in live web applications through statistical inference.

Think of a web application as a black box, with its inputs as HTTPrequests and its outputs as HTTP response data. After a client instruments her web app with our javascript, Taazr will collect request/response data to build a model of expected behavior. When an "anomalous" response is detected, we notify our client, sending along details about the potentially faulty request/response pair.

Don't be vague. Going into detail may seem to leave out the "bigger picture," but I suspect this matters little to the readers at YC. You have the rest of your application to impart a broader vision; for this question, talk specifically about what you are going to build.

Please tell us in one or two sentences about the most impressive thing other than this startup that each founder has built or achieved.

It's often said that YC is all about the founders. Our answer:

Ethan was admitted to CS PhD programs at Stanford, Berkeley, and MIT for research on *automatic software development (e.g. automatically finding bugs, fixing bugs, generating test suites).*

Muzzammil is a member of the core development team for GuardRails, a secure web application framework, which will be published at USENIX 2011 and presented at RubyNation. He is also the founder of Wahoobooks, a site where U.Va. students can list used textbooks.

We mentioned specific and verifiable achievements, the substance of which implied that we are determined people. You should try to avoid generalities.

Please tell us about the time you most successfully hacked some (non-computer) system to your advantage.

A great answer here can really help you, but it was perhaps our weakest response. Judge for yourself:

A Computer Science degree from U.Va. typically requires that you complete many courses within a variety of unrelated disciplines. Most of our peers must take classes in Physics, Chemistry, and a "fake" humanities listing called Science, Technology and Society. We escaped such requirements through a little-known option to pursue a BA in Computer Science, in place of the Engineering School's BS. With this degree, we completed the core CS curriculum and any other classes that caught our interest, but we avoided the rigidly defined ABET requirements of the Engineering school. This left us more time to pursue research and hack on web applications.

While Muzzammil and I aren't totally straight-laced, we couldn't think of much to say. And that was a strategic error, for a sufficiently devious answer has a great opportunity to stand out. While I like to think that the most clever subversives are the overtly boring ones, that's not correct so far as this question is concerned. Give a better answer than we did!

Please tell us about an interesting project, preferably outside of class or work, that two or more of you created together. Include urls if possible.

I'd rate our answer here as about average. We'd worked together before, but not on anything terribly impressive.

For a mobile development class, we created a Barefoot Running application[1] *for the iPhone. The app allows users to report and view the location of glass shards – centralized in an off-site database – and keeps track of speed and distance data throughout a run.*

We also jointly created BijectKarma, a community connecting designers and developers[2]*. We had trouble attracting initial users, however, and the site was taken offline.*

Perhaps I should give a plug to barefoot running, as clearly that's the reason we were accepted. Go work on cool things.

How long have the founders known one another and how did you meet? Have any of the founders not met in person?

This one is easy to interpret. Come application time, there's nothing much you can do to improve your answer.

1.5 years. Ethan was Muzzammil's TA.

While working on a problem set, Muzzammil mentioned his interest in startups. Ethan asked if he had heard of Y Combinator, which of course Muzzammil had. As it turned out, we were both fishing for potential co-founders.

So yes, you should meet people sooner, rather than later. It's better if you can convey that your founders know each other well.

Why did you pick this idea to work on? Do you have domain expertise in this area? How do you know people need what you're making?

I like our answer to this one, but with a few caveats.

Finding and fixing software bugs is expensive. Taazr will save companies time and money by identifying bugs and potential fixes in web applications. We chose to work on Taazr primarily due to our familiarly with related research fields, but also because it represents a challenging problem.

Ethan has two years of experience hacking with the Automatic Program Repair (APR) research group at U.Va, where he's worked with state-of-the-art techniques in program analysis, testing, and statistical debugging. He has published work increasing the efficiency and scalability of APR.

Muzzammil has a year of experience with the GuardRails research group at UVA, where he helped create a secure web framework for Ruby on Rails. His work is published in USENIX 2011 and will also be presented at RubyNation.

More generally, Taazr was inspired by the Cooperative Bug Isolation Project[2]*, in combination with ideas gleaned from our respective research groups at U.Va.*

We describe our domain expertise, and slip in evidence that we can get things done. However, we do come off as a bit research-y, which I think is a bit of a risk. Seek to convey expertise through specific accomplishments. You should be perceived as people who know what they are getting into.

What's new about what you're making? What substitutes do people resort to because it doesn't exist yet (or they don't know about it)?

For us, this was almost a continuation of the previous question.

Uncaught program bugs are expensive, and for this reason, good developers take care in testing their code. However, even the best test suites will not catch all program bugs. Taazr will identify such bugs earlier in the development cycle, and companies will save money that otherwise might have been spent on code maintenance and customer support.

We believe that statistical debugging is uniquely suited to web applications, due to the relative ease with which web apps can be instrumented through lightweight and unobtrusive javascript. While statistical debugging has succeeded in a research context on desktop software, automatically finding bugs in live web applications remains an open industrial problem.

However, our ultimate aim is not just to find bugs in web applications, but also to fix them. This will likely require tighter integration – server-side – between our tools and a developer's code.

I'd once more levy the "research-y" criticism against our response. An answer to this question will depend upon your product, and I doubt it matters whether your idea is truly novel. As always, clarity is important.

Who are your competitors, and who might become competitors? Who do you fear most?

Another straightforward one.

Coverity and Klocwork also find program bugs, but they don't target web applications. Moreover, engineers use their products throughout the development process, whereas Taazr operates, automatically, on live production code.

It's worth noting that we didn't fully answer this question. I suppose that's because we are fearless, but I'd recommend a different approach.

What do you understand about your business that other companies in it just don't get?

Our answer here was weird.

Writing code is much like putting words to paper, an act of creation. When you write, it is important to spell things correctly, but few writers spend much time on this task, for they have spellcheckers. We think that testing should be just as straightforward. It is quite important, but should not monopolize your attention.

Our response is both wrong – I now disagree with it – and confusing. Testing is a bit of a non-sequitur. You should give a real reason why you have insight over your competitors.

How do or will you make money? How much could you make? (We realize you can't know precisely, but give your best estimate.)

We had no idea how much we could make.

Taazr will charge clients monthly. We see a market opportunity in the tens of millions.

I'm not sure this question matters much, as it's pretty easy to see how most products might become profitable. If your idea is particularly eccentric, then perhaps a good answer matters more.

Please tell us something surprising or amusing that one of you has discovered. (The answer need not be related to your project.)

This is a fun question to answer, but I don't think it heavily impacts your application.

You can start small word-memes among family and friends. Choose a word that is fairly obscure (but not obviously so), something like pedantic, sycophant or obsequious. Over a few days, casually inject it into conversation.

Listen throughout the next week. The word will come up with surprising frequency. You can also do this with unique inflections, and distinct rhythms of speech.

You should try to be interesting and clever, I suppose.

The rest of the questions?

Our responses were one-liners which suggested that we had begun market research, but didn't have any customers. Nothing insightful. But we did receive a follow-up question from Robert Morris...

Robert Morris: Why should I believe that this will work?

We gave him a well-grounded but complicated answer, as suits an MIT Professor.

The best evidence probably comes from recent web security research. For instance, Vigna et al. has successfully used anomaly detection to identify security vulnerabilities in web applications. While we are attacking a different goal (identifying bugs, not vulnerabilities), our current work borrows heavily from techniques used successfully in anomaly detection systems.

Since our models will be trained on more detailed user metrics than has traditionally been attempted (e.g. user mouse movements and clicks, as opposed to just HTTP request strings), we see the real question as whether we can distinguish between anomalous user activity (which may signify a bug, or some other issue) and the noise which you would expect in user-based data.

We didn’t receive a follow-up question, although it’s unclear whether that’s a good thing.

Concluding thoughts

I hope you found this useful. It was fun to dig through our old application, particularly as we have since gone through a full pivot: Taazr became Proxino. Things change; such is the nature of time, and confidant YC applications.

As final advice, I would focus on brevity and concision in your answers. The people at YC read through quite a lot of these, after all. I’d also keep in mind that ideas are fungible, and people are not. YC cares about the founders.

And if you take away anything from this post, it should be that groups with imperfect applications can get into Y Combinator. So be inspired. Best of luck to applicants for the winter batch.

A Successful YC Application: Dropbox

Y Combinator Funding Application, Summer 2007

Application deadline: 12 midnight (PST) April 2, 2007. Please try to answer each question in less than 120 words.

We look at online demos only for the most promising applications, so don’t skimp on the application because you’re relying on a good demo. We don’t make any formal promise about secrecy, but we don’t plan to let anyone outside Y Combinator see these applications, including other startups we fund. We recommend you save regularly by clicking on the update button at the bottom of this page. Otherwise you may lose work if we restart the server.

Username:

dhouston

Company name:

Dropbox

Company url, if any:

http://www.getdropbox.com/

Phone number (preferably cell):

(redacted)

# Usernames of all founders, separated by spaces. (Please have all founders create YC accounts, or

create accounts for them.) *

dhouston

Usernames of all founders who will move to (or already live in) Boston for the summer if we fund you.

dhouston

What is your company going to make?

Dropbox synchronizes files across your/your team's computers. It's much better than uploading or email, because it's automatic, integrated into Windows, and fits into the way you already work. There's also a web interface, and the files are securely backed up to Amazon S3. Dropbox is kind of like taking the best elements of subversion, trac and rsync and making them "just work" for the average individual or team. Hackers have access to these tools, but normal people don't.

There are lots of interesting possible features. One is syncing Google Docs/Spreadsheets (or other office web apps) to local .doc and .xls files for offline access, which would be strategically important as few web apps deal with the offline problem.

For each founder, please list:

YC username; name; age; year, school, degree, and subject for each degree; email address; personal url (if any); and present employer and title (if any). Put unfinished degrees in parens. List the main contact first. Separate founders with blank lines. Put an asterisk before the name of anyone not able to move to Boston for the summer.

dhouston; Drew Houston; 24; 2006, MIT, SB computer science; Houston AT alum DOT (school I went to) DOT edu; –; Bit9, Inc (went full time to part time 1/07) – project lead/software engineer

Although I'm working with other people on Dropbox, strictly speaking I'm the only founder right now.

My friend (redacted), a great hacker, Stanford grad and creator of (redacted) is putting together a Mac port, but can't join as a founder right now as a former cofounder of his started an extremely similar company.

My friend and roommate (redacted) from MIT is helping out too, but he works with me at Bit9, and a non-solicit clause in my employment contract prevents me from recruiting him (and the VP Eng explicitly told me not to recruit him.)

In any case, I have several leads, have been networking aggressively, and fully intend to get someone else on board – either another good hacker or a more sales-oriented guy (e.g. the role Matt fills at Xobni). I'm aware that the odds aren't good for single founders, and would rather work with other people anyway.

Please tell us in one or two sentences something about each founder that shows a high level of ability.

Drew – Programming since age 5; startups since age 14; 1600 on SAT; started profitable online SAT prep company in college (accoladeprep.com). For fun last summer reverse engineered the software on a number of poker sites and wrote a real-money playing poker bot (it was about break-even; see screenshot url later in the app.)

What's new about what you're doing?

Most small teams have a few basic needs: (1) team members need their important stuff in front of them wherever they are, (2) everyone needs to be working on the latest version of a given document (and ideally can track what's changed), (3) and team data needs to be protected from disaster. There are sync tools (e.g. beinsync, Foldershare), there are backup tools (Carbonite, Mozy), and there are web uploading/publishing tools (box.net, etc.), but there's no good integrated solution.

Dropbox solves all these needs, and doesn't need configuration or babysitting. Put another way, it takes concepts that are proven winners from the dev community (version control, changelogs/trac, rsync, etc.) and puts them in a package that my little sister can figure out (she uses Dropbox to keep track of her high school term papers, and doesn't need to burn CDs or carry USB sticks anymore.)

At a higher level, online storage and local disks are big and cheap. But the internet links in between have been and will continue to be slow in comparison. In "the future", you won't have to move your data around manually. The concept that I'm most excited about is that the core technology in Dropbox – continuous efficient sync with compression and binary diffs – is what will get us there.

What do you understand about your business that other companies in it just don't get?

Competing products work at the wrong layer of abstraction and/or force the user to constantly think and do things. The "online disk drive" abstraction sucks, because you can't work offline and the OS support is extremely brittle. Anything that depends on manual emailing/uploading (i.e. anything web-based) is a non-starter, because it's basically doing version control in your head. But virtually all competing services involve one or the other.

With Dropbox, you hit "Save", as you normally would, and everything just works, even with large files (thanks to binary diffs).

What are people forced to do now because what you plan to make doesn't exist yet?

Email themselves attachments. Upload stuff to online storage sites or use online drives like Xdrive, which don't work on planes. Carry around USB drives, which can be lost, stolen, or break/get bad sectors. Waste time revising the wrong versions of given documents, resulting in Frankendocuments that contain some changes but lose others. My friend Reuben is switching his financial consulting company from a PHP-based CMS to a beta of Dropbox because all they used it for was file sharing. Techies often hack together brittle solutions involving web hosting, rsync, and cron jobs, or entertaining abominations such as those listed in this slashdot article (Small Office Windows Backup Software).

How will you make money?

The current plan is a freemium approach, where we give away free 1GB accounts and charge for additional storage (maybe ~$5/mo or less for 10GB for individuals and team plans that start at maybe $20/mo.). It's hard to get consumers to pay for things, but fortunately small/medium businesses already pay for solutions that are subsets of what Dropbox does and are harder to use. There will be tiered pricing for business accounts (upper tiers will retain more older versions of documents, have branded extranets for secure file sharing with clients/partners, etc., and an 'enterprise' plan that features, well, a really high price.)

I've already been approached by potential partners/customers asking for an API to programmatically create Dropboxes (e.g. to handle file sharing for Assembla.com, a web site for managing global dev teams). There's a natural synergy between Basecamp-like project mgmt/groupware web apps (for the to-do lists, calendaring, etc.) and Dropbox for file sharing. I've also had requests for an enterprise version that would sit on a company's network (as opposed to my S3 store) for which I could probably charge a lot.

Who are your competitors, and who might become competitors? Who do you fear most?

Carbonite and Mozy do a good job with hassle-free backup, and a move into sync would make sense. Sharpcast (venture funded) announced a similar app called Hummingbird, but according to (redacted) they're taking an extraordinarily difficult approach involving NT kernel drivers. Google's coming out with GDrive at some point. Microsoft's Groove does sync and is part of Office 2007, but is very heavyweight and doesn't include any of the web stuff or backup. There are apps like Omnidrive and Titanize but the implementations are buggy or have bad UIs.

For founders who are hackers: what cool things have you built? (Include urls if possible.)

Accolade Online SAT prep (launched in 2004); a poker bot (the link is to an old screenshot; it's using play money there but worked with real money too.)

How long have the founders known one another and how did you meet?

There's a joke in here somewhere.

What tools will you use to build your product?

Python (top to bottom.) sqlite (client), mysql (server). Turbogears (at least until it won't scale.) Amazon EC2 and S3 for serving file data.

If you've already started working on it, how long have you been working and how many lines of code (if applicable) have you written?

3 months part time. About ~5KLOC client and ~2KLOC server of python, C++, Cheetah templates, installer scripts, etc.

If you have an online demo, what's the url?

Here's a screencast that I'll also put up on news.yc.

If you do have a Windows box or two, here's the latest build.

How long will it take before you have a prototype? A beta? A version you can charge for?

Prototype – done in Feb. Version I can charge for: 8 weeks maybe? (ed: hahaha)

Which companies would be most likely to buy you?

Google/MS/Yahoo are all acutely interested in this general space. Google announced GDrive/"Platypus" a long time ago but the release date is uncertain (a friend at Google says the first implementation was this ghetto VBScript/Java thing for internal use only). MS announced Live Drive and bought Foldershare in '05 which does a subset of what Dropbox does. Iron Mountain, Carbonite or Mozy or anyone else dealing with backup for SMBs could also be interested, as none of them have touched the sync problem to date.

In some ways, Dropbox is for arbitrary files what Basecamp is for lightweight project management, and the two would plug together really well (although 37signals doesn't seem like the buying-companies type).

At the end of the day, though, it's an extremely capital-efficient business. We know people are willing to pay for this and just want to put together something that rocks and get it in front of as many people as possible.

If one wanted to buy you three months in (August 2007), what's the lowest offer you'd take?

I'd rather see the idea through, but I'd probably have a hard time turning down $1m after taxes for 6 months of work.

Why would your project be hard for someone else to duplicate?

This idea requires executing well in several somewhat orthogonal directions, and missteps in any torpedo the entire product.

For example, there's an academic/theoretical component: designing the protocol and app to behave consistently/recoverably when any power or ethernet cord in the chain could pop out at any time. There's a gross Win32 integration piece (ditto for a Mac port). There's a mostly Linux/Unix-oriented operations/sysadmin and scalability piece. Then there's the web design and UX piece to make things simple and sexy. Most of these hats are pretty different, and if executing in all these directions was easy, a good product/service would already exist.

Do you have any ideas you consider patentable?

(redacted)

What might go wrong? (This is a test of imagination, not confidence.)

Google might finally unleash GDrive and steal a lot of Dropbox's thunder (especially if this takes place before launch.) In general, the online storage space is extremely noisy, so being marginally better isn't good enough; there has to be a leap in value worthy of writing/blogging/telling friends about. I'll need to bring on cofounder(s) and build a team, which takes time. Other competitors are much better funded; we might need to raise working capital to accelerate growth. There will be the usual growing pains scaling and finding bottlenecks (although I've provisioned load balanced, high availability web apps before.) Acquiring small business customers might be more expensive/take longer than hoped. Prioritizing features and choosing the right market segments to tackle will be hard. Getting love from early adopters will be important, but getting distracted by/releasing late due to frivolous feature requests could be fatal.

If you're already incorporated, when were you? Who are the shareholders and what percent does each own? If you've had funding, how much, at what valuation(s)?

Not incorporated

If you're not incorporated yet, please list the percent of the company you plan to give each founder, and anyone else you plan to give stock to. (This question is as much for you as us.)

Drew

If you'll have any major expenses beyond the living costs of your founders, bandwidth, and servers, what will they be?None; maybe AdWords.

If by August your startup seems to have a significant (say 20%) chance of making you rich, which of the founders would commit to working on it full-time for the next several years?

Drew

Do any founders have other commitments between June and August 2007 inclusive?

No; I've given notice at Bit9 to work on this full-time regardless of YC funding.

Do any founders have commitments in the future (e.g. have been accepted to grad school), and if so what?

No. Probably moving to SF in September

Are any of the founders covered by noncompetes or intellectual property agreements that overlap with your project? Will any be working as employees or consultants for anyone else? *

Drew: Some work was done at the Bit9 office; I consulted an attorney and have a signed letter indicating Bit9 has no stake/ownership of any kind in Dropbox

Was any of your code written by someone who is not one of your founders? If so, how can you safely use it? (Open source is ok of course.) *

No

If you had any other ideas you considered applying with, feel free to list them. One may be something we've been waiting for.

One-click screen sharing (already done pretty well by Glance); a wiki with version-controlled drawing canvases that let you draw diagrams or mock up UIs (Thinkature is kind of related, but this is more text with canvases interspersed than a shared whiteboard) to help teams get on the same page and spec things out better (we use Visio and Powerpoint at Bit9, which sucks)

Please tell us something surprising or amusing that one of you has discovered. (The answer need not be related to your project.)

The ridiculous things people name their documents to do versioning, like “proposal v2 good revised NEW 11-15-06.doc”, continue to crack me up.

XIV.

Chapter 13: Inside the Interview

Stories And Advice: Mailgun – Earbits

YC Interview Stories and Advice from 8 Companies in YCW11

When we blogged about a YC application checklist, it was fairly easy to compile because a successful app is more dependent on people than the company. Moreover, the GiftRocket team had a reviewed a ton of applications, so we had an understanding of what made some good and others not as good.

Interviews are different. We've only had/seen one. So we called in some favors from our friends in the YCW11 class and compiled the stories of 8 companies: Mailgun, Like.fm, Grubwithus, Earbits, LAL, Zerocater, Tutorspree and GiftRocket.

We asked each company five questions:

- **Traction: Where were you at in the process when you interviewed?**
- **Content: After you explained what your company did, what did the partners ask about?**
- **What they liked: What did they find most impressive about you?**
- **Hard questions: Did they ask anything that flustered you?**
- **Advice: What advice would you give to companies going through the interview process?**

The answers are mostly verbatim, with some light editing for readability.

Mailgun

Mailgun is an easy email service for developers.

Traction: When we interviewed, Mailgun only had a few early customers. We missed bells and whistles like automated billing, and the sign-up process was not 100% automated, but we could sign up customers manually and could demonstrate that the product was making people happy.

Content: We were lucky enough to relate directly to previously frustrating experiences for YC partners. The first thing PB said when we entered the room was “We could have used you guys at FriendFeed.” It helped that everyone at the table was a programmer and the pain of dealing with email was something everyone could relate to. Email is a huge problem which stretches well beyond sending. Nobody disagreed with that. PB, being the father of Gmail, grilled us a bit on the specifics of email business to make sure we’re weren’t making this up, and that was it.

What they liked: They didn’t comment much on our personas, but I vividly remember PG getting super excited about the possibility of email-based RPC and giving each function in a program its own email address.

Hard questions: For some reason we were lucky enough to get only technical questions, i.e. they were probing for bugs. And we had “unit tests” for that: we knew the files inside out and made it very apparent right from the start.

Advice: Trying to prove something, it is often easier to have a single but amazing and irrefutable argument than a bunch of so-so ones. Walking into that room, it helps to have a single killer answer to this one question: “why will we succeed?” And no matter what they ask, just go with it, because that’s really the answer they’re looking for regardless of the shape and form of the question on the table. Let them know *we’re doing $100K/month and growing 120% week to week*. Don’t hope for them to stumble upon your greatness, show it off yourself.

I think one of the most important things you can do to help your case is to have a working product (no matter how poor it is relative to your vision) and have evidence that people are benefiting from it. If they are paying for it, even better. Remember, the motto is *make something people want*. If you don’t have this, you had better be an expert at what you are building and be able to evidence this.

Like.fm

Like.fm is a music recommendation service that works by automatically mining data on what you play.

Traction: I had only about 10k users when I was interviewing – that’s not enough traction to justify ignoring all other factors. I think it’s a big enough number to show potential, not necessarily for the product, but that I can at least get something floating above ground.

Content and hard questions: I forgot what they asked me right afterwards. I think it was some stuff about why Last.fm users would switch.

What they liked: They seemed to like that I was going after a big thing (Last.fm and their millions of users). They did ask something that I wasn't expecting, but I forgot what it was. Might have been why Last.fm users would switch over.

Advice: There are things you do that lets them know you really are committed, all-in, focused, and determined to succeed. Do those things, and make sure the partners know about those qualities before the interview is over.

Grubwithus

Grubwithus is a site that helps you meet people over meals. They've talked about getting into YC on their blog, and are holding meals with founders for the upcoming batches.

Traction and content: When we interviewed, we had just launched in one city: Chicago. They asked how we expected to get traction for a marketplace business.

What they liked: They were most impressed by how much hustle we had…and how we had known each other for years and had worked together on a ton of stuff.

Hard questions: They didn't ask many questions that flustered us…the interview went really smoothly, actually, and we were pretty sure we got in after the interview.

Advice: Smile and be likable. YC is not just a 3 month program; the partners have to deal with you for the rest of your life, so they're much more likely to fund people they get along with.

Earbits

Earbits is commercial-free radio for indie music. They've talked about getting into YC before, here.

Traction: When we applied, we had a beta site up, $110k in friends and family funding, 5 or 6 on the team, and 30-40 partners we were working with. We didn't have any revenue.

Content: It's all kind of a blur. They asked questions so fast that you often only got to say the first few words of the answer before you were onto something else, but that was generally a good thing because it meant they were excited and wanted to learn more. Mostly they asked why we were doing what we were doing, and how we were approaching trouble spots in our business.

What they liked: I believe what they found most intriguing about us was that we were approaching a very typical business with a completely new model and distribution approach. It was not some small adjustment to a typical business model – we were really revolutionizing the way a music streaming service would work and we weren't scared of it. We knew why we wanted to do it this way, it was based on real experience, and we had a team with the perfect background to do it.

Hard questions: They didn't ask anything that flustered us, but they kept asking question after question non-stop, and we really just wanted them to see what we had already built. They never asked to see it; we had to make it a point to show them. So, we kept answering their questions with, "Let us show you the product." After saying that two or three times, we finally managed to turn the focus to the demo, and it was very valuable. If you have a good product or demo, make sure you show it off – and don't expect them to ask you to do so. But if you're going to show them something, make damn sure it works.

Advice: Chill out, have fun with it. If you have what YC is looking for, it's going to come out best if you go in relaxed; just engage in the interview like you're chatting with fun people who love hearing new ideas. Be very flexible about how you think about your business. Answer their questions with your thoughts and back up firm opinions with data, but be ready to say, "that's a good idea" or "that may work." These guys have seen a million ideas. If you know something about your business for certain, you can tell them why, but if you don't, you can expect that they'll have valuable insights.

Stories And Advice: LAL – GiftRocket

LAL (LikeALittle)

LAL is a flirting site that has taken college campuses by storm.

Traction and content: When we applied, we had an initial product and a good amount of traction. They asked us about the things we've learned so far from the current product (although I don't recall exactly).

What they liked: They were impressed by our team (backgrounds and a history of perseverance), willingness to do groundwork to get users and user feedback, and initial traction.

Hard questions: They didn't ask anything that flustered us.

Advice: First off, show that your team is dedicated to doing whatever it takes to be successful. They know the number one reason startups fail is because they give up or stop trying so take that doubt out of their mind entirely. Number two is being extremely scrappy in talking to/listening to users on a consistent basis. Every week, there should be touch points with users and new insights about the product from users. Show that you're doing that often.

Zerocater

Zerocater helps companies feed their employees.

Traction and content: When we applied, we had revenue on a very minimal product. We'd heard stories about how hard the interviews were, but we were lucky to have an easier time than others. It took us three minutes to explain the idea, and for the rest of the time, Paul was brainstorming ideas for us.

What they liked: We were solving a problem they'd experienced firsthand. Paul asked Jessica how she felt about organizing food for Y Combinator events. Her answer was an emphatic "I hate it!"

Hard questions: We were generally prepared for all of the questions they asked.

Advice: Traction forgives all. It helped us have an easier time with the interview.

Tutorspree

Tutorspree is an online marketplace for private tutors.

Traction: We hadn't launched yet, but we had managed to sign up a couple hundred tutors, mostly in the NYC area. We had no paying customers at this time.

Content: They asked a lot about the size of our market, and how we differed compared to any existing competitors. We thought we were much better and gave them the reasons why. Tutoring is an enormous market domestically here in the U.S., about $7B.

What they liked: I (Ryan) had startup experience at SeatGeek which I think they liked – and also Josh was able to say he was the #1 diamond dealer on Amazon at the time. This was true, and they all thought it was cool and got a laugh out of it.

Hard questions: They asked how much customer acquisition cost us – we hadn't experimented too much with PPC yet, so we told them we weren't sure yet. We probably should have had an answer for this though.

Advice: Have a friend pretend to be PG and drill you with questions about your company. We did this and it seemed to be very helpful.

GiftRocket

GiftRocket is an online gift card company.

Traction: When we interviewed, we had just switched to a new idea from what we wrote on our application. We had discussed it briefly with Harj in one of the video chats.

Content: They first asked us about the market, and whether people would be willing to use this. We had done extensive research here and told them about our findings.

What they liked: They liked our idea a lot; they liked that we were going after something massive; they liked that we were willing to switch to an entirely new idea than what we applied with; they liked that we all wrote code and had the beginnings of a website/app within a short timeframe; they liked that we had already quit our jobs and were going to be working on this either way.

Hard questions: They asked about managing fraud and about how we'd market the product. We hadn't thought through either. The marketing question was excusable since it was too early for us to run tests, and PG himself defended us on that point. The fraud one was less excusable and we rattled off some half-baked answer that wasn't too impressive.

Advice: If you're early, know your market inside out. They asked us a few questions about market size, smartphone usage in the US, growth, and so on. We had the precise answers and communicated them with full confidence despite the fact we had come up with the idea days ago. As well, have something to show, even if its just wireframes. We had already started working on the website and an app. The graphics were extremely basic but it helped show that we were able to get stuff done.

XV.

Sources

An Epic List Of Links

Links Used in this Ebook

- How To Apply to Y Combinator [PG]
- How To Get Into Y Combinator [Sam Altman]
- Relentlessly Resourceful [PG]
- How to Be Relentlessly Resourceful [Jason Shen]
- What We Look for in Founders [PG]
- The Anatomy of Determination [PG]
- 18 Mistakes that Kill Startups [PG]
- Super Last Minute Advice For Startups Applying For Y Combinator [Garry Tan, Posterous]
- How Should I Fill Out the Application Form for Y Combinator? [Quora]
- Directed Edge HN video

Additional Reading

- So You Want to Get Into Y Combinator?
- Opzi’s Euwyn Shares What You’ll Get Out Of Y Combinator [Maple Butter]
- How Not to Pitch to Y Cominator [Damien Katz]
- How to Get Into Y Combinator [Christopher Golda]
- How Y Combinator Helped 172 Startups Take Off [Mixergy]
- Our Y Combinator Experience [Upverter]
- What are the chances for a non-technical single founder startup applying to Y Combinator? [Quora]
- The Value of the Y Combinator Experience [Sachin Rekhi]
- Late applications [Hacker News]
- Does Going Through an Incubator Send a Negative Signal To VCs? [Quora]
- GiftRocket Application Checklist
- Chances For a Non-Technical Founder Startup [Quora]
- Best Advice For a Startup Applying to Y Combinator [Quora]

- GiftRocket Interview Prep
- What Are Your Chances With YC? [Hacker News]

Which Friends Will Receive Your 5 Bonus Books?

Hyperink Bonus — As a bonus for purchasing a *Hyperink* book, you now have **5 bonus copies to share with your friends**, for free!

Click Here to Choose your 5 friends!

We're offering a special bonus to Hyperink customers who purchase special titles:

Go to https://hyperink.wufoo.com/forms/gift-this-book-to-5-of-your-friends/def/Field17=Guide+To+YC, and send up to 5 bonus copies of the **Guide To YC** to your friends and family! If you've ever thought about starting your own book club–now is the time.

Did you receive the book as a gift from a friend or family member? You can still take advantage of the offer - we want to share our books with as many people as possible!

In the following 1-minute form, simply fill out your email, and the emails of *up to 5 people* that you'd like to receive your gift.

They'll receive a unique download link for a free copy of the **Guide To YC**.

We take your privacy very seriously, and we promise that we will NEVER spam you or share your email with any 3rd parties.

Please email hello@hyperink.com if you have any questions or feedback.

Thanks, and enjoy the book!

-The Hyperink Team

Facebook: www.facebook.com/hyperink
Twitter: www.twitter.com/hyperink

About The Publisher

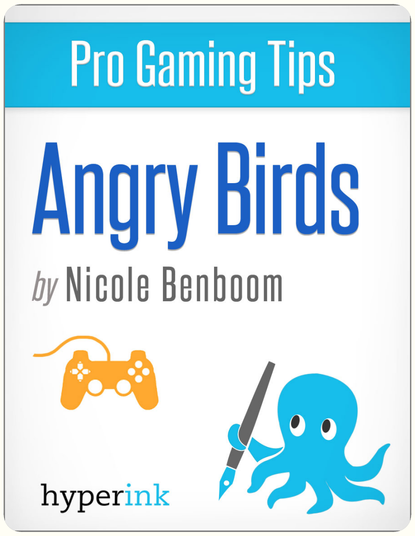
Hyperink is the easiest way for anyone to publish a beautiful, high-quality book.

We work closely with subject matter experts to create each book. We cover topics ranging from higher education to job recruiting, from Android apps marketing to barefoot running.

If you have interesting knowledge that people are willing to pay for, especially if you've already produced content on the topic, please reach out to us! There's no writing required and it's a unique opportunity to build your own brand and earn royalties.

Hyperink is based in SF and actively hiring people who want to shape publishing's future. Email us if you'd like to meet our team!

Our Readers' Favorite Books



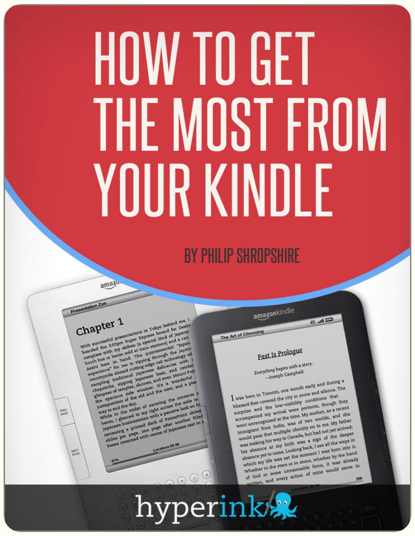
The Best Book On Pro Gaming Tips: Angry Birds

Nicole Bemboom

Your Guide To Gaming Well.

25% off Purchase this Book →

25% off the original price. Limited time offer.



How To Get The Most From Your Kindle Device

Philip Shropshire

Hidden tips and tricks for your favorite eReader!

25% off Purchase this Book →

25% off the original price. Limited time offer.

Our note:

Please don't make copies of this book. We work hard to provide the highest quality content possible - and we share a lot of it for free on our sites - but these books are how we support our authors and the whole enterprise. You're welcome to borrow (reasonable) pieces of it as needed, as long as you give us credit.

Thanks!

The Hyperink Team

Disclaimer

This ebook provides information that you read and use at your own risk.

We do not take responsibility for any misfortune that may happen, either directly or indirectly, from reading and applying the information contained and/or referenced in this ebook.

This Quicklet, Best Book, and Hyperink are not affiliated with or sponsored by the original work or its author or publisher(s). This Quicklet and Best Book are intended to provide educational commentary and analysis about the original work, but this Quicklet and Best Book are not endorsed by the author or publisher(s) of the original work.

Thanks for understanding. Good luck!