

The logo consists of the lowercase letters 'bn' in a white, bold, sans-serif font, set against a solid red square background.

BlogNewcomb

Cult Creation

Not all cults are bad. In fact for startups they can make all the difference. Learn how to build a team from one person to an unstoppable army.



Cult Creation

Over the years, I've been accused by the press of many things, but one of my favorites is the articles that claim that I create cults or even religious orders in my companies. While you'd think I might be pissed, there's a lot of truth in those articles. In fact, I'm quite proud of being recognized for creating cult-like cultures in my companies, because it's actually part of my plan.

Why? If you are a founder of a startup, then burn the following info into head.

Having a good team, or even an excellent team isn't enough - you

need to build a "cult." And by "cult" what I mean is a group of super high quality people who trust each other and have similar ways of thinking, learning, reacting, problem-solving and working together. Further, this team needs to bond together under a leader they trust and respect. They

must not be afraid of any type of challenge; they must be willing to walk through fire when their leader asks them; and they must believe they deserve to be the ones that will change the world.

Do this and you and your team can handle anything. Don't, and you'll be heading to the dead pool.

By "cult" what I mean is a group of super high quality people who trust each other and have similar ways of thinking, learning, reacting, problem-solving and working together

As a founder, I can tell you that one of the very few things that will enable you to get some sleep at night is knowing that you're not the leader of just a company, but rather a cult following.

If you're like nearly every founder I know, you worry a lot.

What happens when founders try to sleep at night is this: their mind spins uncontrollably between scenarios that result in a glorious



success and scenarios that result in a burning death crash. Most founders I know ride a fine line between seeming to be in total control and going nut balls.

All I can say is welcome to the club - you're normal!

Whenever people ask me how I make it through, I always say the same thing. Sit down and write down the shit storms that you are worrying about and divide them into two lists. Those that are under your control and those that aren't. Then focus on the list that you can control. If you stare at that list long enough you'll realize a commonality. That the solution to every single one of them begins with having a team that is rock solid, one that isn't afraid of challenges and one that believes in you as a founder. If you do this one thing right, it will steady you and calm your mind enough to face and conquer any challenge.

If you don't do this one thing right, it won't really matter how well you do anything else. Team is the genesis of all success and if you don't nail it you're not going to get much sleep. Fuck up your team, and no matter how much you tell yourself otherwise, you're fucked!

So how do you build a cult? Although Gawker would lead you to believe that it may involve getting naked and killing a goat, it doesn't.

Creating a cult starts with the way in which you build your team - *the hiring process*. If you build your team in a certain way, the culture inherent in that team will give birth to the cult itself. It won't be you, as the founder, who will own or control the credo of the cult, it will be the team itself. Your job is to control the processes by which the team is built. Implement processes that are poorly thought out and you'll create a cult of losers. Nail it, and your team will create their own cult of killer employees.

The intent of this essay is to gather in one place the principles and the practices that I have learned over my own many years of sometimes doing it the right way and sometimes fucking it up. These are the golden rules that I now try to utilize in all of my companies. In some cases, the principles are results of lessons that I learned through people that have had a direct influence in my career like Peter Thiel, founder of Paypal, and Bill Keating. In other cases, they are the direct result of practical experience.

Principles

IF THEN Statements Work - developed during the process of me convincing the first 8 employees of Powerset to join.

For many founders, hiring the initial team is one big chicken and egg problem. Investors want to see a group of super talented people committed to your idea before you have any money. Employees, who are much more risk averse than founders, want to see that you already have money, or that your money is nearly assured before joining. It's a common challenge.

So who wins this chicken and egg?

Well I'm a big believer that the best ideas are those where you can convince people to join even when there is no money. Following the talent is an easy way to see where the best startups are forming.

So how do you do this?

OK. So here is what I have learned:

1. Write down the names of your founding employee targets - the initial people you are trying to convince to join.
2. Write down any dependencies that exist between those people. i.e. if that guy joins then I'll join.
3. Write down the dependencies for each person that are independent of the other candidates.
4. Highlight dependencies from #3 that are common among the candidates.
5. Write down those dependencies from #4 that are necessary to solve before your next

conversation with each candidate

6. Write down those dependencies from #4 that you need to solve before they would be willing to sign

an offer letter

7. Write down those dependencies from #4 that you need to solve before each candidate is willing to quit their current job and join -

At Powerset, the first 8 employees actually had IF THEN statements in their employment offers

notice I differentiated between signing an offer letter and quitting their existing job and working full time.

Once you're done this exercise, you should end up with a long, but understandable IF THEN statement.

You should now know the order in which you need to land your team and the order of the dependencies you need to solve. This gives you your "to do" list and what should be the top priorities of the company.

By doing this exercise, you'll also know what dependencies will exist at the time you want them to sign an offer letter. That's critical. Most first time founders think you have to have all dependencies met before you can ask a person to sign an offer letter. You don't - put those dependencies inside the offer letter as a condition of starting employment instead of a start date. In other words, the day those dependencies are solved, they agree to give notice to their existing

company and begin at yours within 2-3 weeks or so.

Once you have signed offer letters, even if you have dependencies in them, investors, partners and other

candidates will see this as a sign of strength. This is often called "lookin like a duck and quackin like a duck" - it shows momentum and that's always critical. It helps you maintain your cadence and always gives you and those that have joined

something to celebrate.

The best way to prove to yourself and to potential investors that you have a killer idea is to see if you can get a number of A level engineers to join you by only offering them equity and no initial salary.

Note: *getting a full salary is generally not a good dependency - if it is, you've got bigger problems with that candidate. Either they really don't believe in your idea, or they are so risk averse you should actually target them in the second or third wave of employee hires and not the first.*

Full-time, Equity-only Commitments are a Key Sign of Strength - from Peter Thiel.

The best way to prove to yourself, potential investors and to any potential future employees that you have a killer idea, is to get a number of A-level engineers to join full-time with equity-only deals.

The reason is simple. While investors, including angels and venture capitalists, are good at picking which markets to go after, they are more often than not in an incredibly poor position to determine whether or not a founder is the real deal, whether the idea is technically doable and whether the founder involved could build the team necessary to succeed.

A-level engineers, on the other hand, are in a great position to know whether the founder is a bullshitter or not, whether the technology is real or not and how doable the whole thing is. If I see A-level

engineers joining any company, I stand up and pay attention - because I know that A-level engineers, unlike founders and many investors, are very risk intolerant and skeptical people by nature. Investors only have to risk capital and a little bit of

time, but engineers considering to join as employees have a much higher threshold to cross. They are risking their livelihoods, their reputations and will be the ones committed to endless hours of working on the thing.

The best way to prove to yourself, potential investors and to any potential future employees that you have a killer idea, is to get a number of A-level engineers to join full-time with equity-only deals.

When a founder convinces A-level engineers to quit and join them full time with equity deals, it is a major indication of the strength of the founder, the technology and of the idea.

Suspend Disbelief, then Think Backwards - from Bill Keating.

Suspend all disbelief, imagine that you are successful, and then work backwards. One of the things I always do with my companies is

outline what core skill areas I need and then what "imaginary" people I would want in each position. I kind of treat it like a "fantasy football league" way of creating what the org chart would look like if we already were a successful company.

It's a good process to follow, because as you are adding key people on board you can compare them to your imaginary best case person. The reason why this is important is because you don't want to get caught in what I call relative positioning.

In all of the mayhem that exists during the early stages of a company, many founders end up only comparing different candidates against each other and forget to remember a constant. The problem is that if you only have B+ level people in your pool, but one is an A-minus, then the A-minus candidate will appear to be the one you want to hire - which would be a mistake. The proper comparison would be to a consistent benchmark of the A+ person you need. Although it's easy

to say that you would never hire an A-minus person in any role, it's easier than you think to accidentally do it. If you put a process in place to ensure that you always compare against an absolute constant imaginary best case scenario, you won't make this mistake.

You can't really tell what someone is like to work with until you work with them. No interviewing technique will ever solve this shortcoming of interviews.

You are only as good as your weakest asshole.

Interviewing is Not Good Enough
- from hiring really important roles too quickly and then regretting my decision one month after I hired the person.

I do not believe in interviewing as a sole means to evaluate the hiring of someone for several reasons.

1. No matter who you are interviewing or what questions you ask (even if they were developed by a PHD psychologist, or Google), I have found that you can't tell how that person is going to gel, or not gel, with other members of your team.



2. Product managers in particular are bitches to hire. They are all but impossible to evaluate in an interview because there is no real test for competence. I have never been able to hire a “good” product person without them going through some trial period.
3. I have found that some people are awesome at interviews but are horrible as employees.
4. I have found some people who are horrible at interviewing have turned out to be the best employees (note: this is generally because many engineers are uncomfortable in social situations).
5. You can never allow everyone that needs to, or even wants to, to interview every candidate - it would simply take too much time out of the day and be too disruptive.
6. Hiring an employee is a pretty permanent thing. It's very hard to fire a person in the U.S. and firing an employee is often a moral shitter for the company because it means you invited someone into the team that didn't fit. That makes it look like the hiring process should have filtered out this person, but didn't. Ultimately

whenever you have to fire someone, employees will almost always feel that you, as the founder, should have divined that this person was an asshole or was unskilled. Everything will always be your fault, so own it.

Hire Slowly, Fire Quickly - from Peter Thiel.

Never, ever make it easy to join your team. In fact, make it very hard to join your team. More talented people will respect this and be excited that it's hard to join your company. It is very important to A-level people that they work with other A-level people. Publicly letting the world know that you have a very difficult hiring process will generally attract more A-level people and will scare off B-level people and below - making your job that much easier.

In terms of firing, remember, if you have 10 people on your team and you have 1 bad team member it can affect your company in many ways - not just because they represent a significant percentage of your company's work capacity.

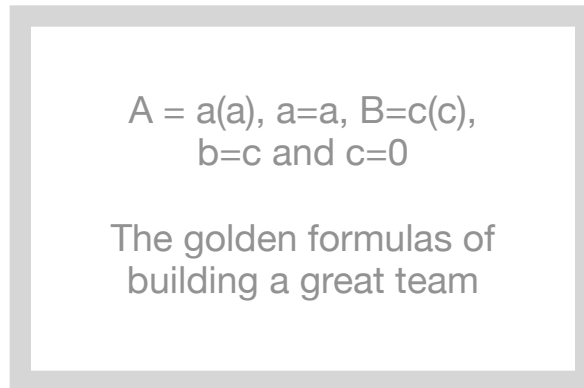
1. If they suck, then their work with suck.

2. It will bum out all of the A-level people that they have to work with a B-level person.

Chances are, if that happens, now you have two problems.

3. It will reduce your credibility with your people if you don't deal with it fast; often times it is painfully obvious to your team that one person needs to go well before you know it. Make sure you stay close to your super stars so that you can act quickly. Staying in touch with your people, acting quickly and acting decisively will improve morale and increase your reputation as a deserving leader.

Super A = a(a), a=a, super B=c(c), b=c and c=0 - Various sources but probably mostly Peter Thiel and a little bit of Luke Nosek.



Super A = a(a) - Uber-connected leaders are super A's. Get one of those and 10 A-level people will join.

a=a. A-level people that aren't super-connected

will at least recruit one other A-level person.

4. Firing quickly can often times dramatically improve morale and bring a renewed sense of confidence; I even got applause once for firing someone.

super B=c(c) Super-connected B-level people, "super Bs," are the most dangerous of all for 2 reasons. People who are B-level know it and therefore often refer C-level people to join so that they end up looking good when compared relative to their co-workers (i.e. b=c).

Therefore Super-Bs will invite 10 C-levels to join, which is a total disaster.

5. This is a big one. If you have someone that needs to go and everyone knows it but you, or you don't act quickly, they may actually refer someone to become an employee and that person may actually get in.

c=0 In an effort to improve the gene pool, C-level people should simply be killed, (metaphorically speaking,

of course) or at least kept as far away as possible from your company.

Your Team Must Have Ownership - from seeing the Powerset cult form in front of me.


As smart as you may think you are, the best way to ensure that you have a high quality team is NOT by micro-managing the situation. Your job is to instill good hiring principles and practices into your team and let them build the team from there.

The easiest way to do that is simply only hire super-A's A-level people, and then let them have control over who is invited to join your cult. What you will find is that they will be potentially even more strict about who can join. The way you know your processes are working is when you overhear employee number 5 saying, "thank God I got a job early on in this company, because there's no way I could get one now."

That would even potentially include you - the founder. The reality of the situation is that the cult is never

really yours, it's your team's cult. They will create it, they will tell you who they want in it. What's more is that they will be right. You will have a shit ton of things on your mind as you build your company. Trust your people and they will build your cult for you.

Make Sure Every Single Candidate is Treated Like Gold - so many startups I know don't do this one, but it's actually pretty big.



You know you're doing it right when employee #5 says
"thank God I got a job early on in this company, because there's no way I could get one now."

Here is the concept. No matter what, every single person that ever interviews, or for that matter touches your company in any way, should have a great experience, regardless of the outcome of the interview.

Here's why.

People talk, text, twitter and update their status on FB and word about how you treat people WILL get around.



Once I know from other people in the company that it is going to be a "no" for someone, I switch gears from hiring to "successful exiting of the candidate." The goal is speeding up the process of getting that person out of the building so it prevents wasting the time of the engineers while making sure that person is respected and treated just as well as anyone who a "yes."

I always pinch myself to remind me that it is critical that even when we want to end the interview immediately (which happens) we do so in a way that is honest, respectful and mindful that we want that person to go away feeling like maybe they are bummed the interview didn't work out, but that the company rocked.

If you do this, believe me it will get around and it will benefit you in many ways. If you don't do this and you act like a pompous ass in the interview, it will get around even faster and it will cripple your ability to hire good people.

I bet right now, off the top of your head, you could name at least one company that has a reputation for treating candidates or its people like shit. That news travels fast and far.

I know that I have jokingly said that C-level candidates should be killed, but in all seriousness, when any candidate comes into your office, make sure your shit is together, you're organized and you take extra time to give respect to the people you say "no" to. If you do say "no," tell them exactly why you are saying "no." Don't bullshit them and don't simply blow them off... it will come back to bite you. One of your key evaluation metrics for your HR person is what percentage of people come away from the interviewing process loving the company. A second one is verifying that every single applicant and interview candidate has been responded to.

Engineers Suck at Negotiating, so Don't Negotiate, Be Fair - from me, after being pissed off about hiring practices I experienced from bad founders.

Over the years, I have noticed some sort of weird inverse correlation between the talent an engineer has for coding and their ability to negotiate. I've seen people that could have hacked into NSA suddenly shit twinkies the second I bring up the topic of their salary. I don't exactly get it, but it's there.



Founders, on the other hand, are almost by nature programmed to negotiate everything. In some cases, I have seen founders take advantage of engineers who don't negotiate well, or who simply hate to negotiate so much that it's a near-phobic experience.

DO NOT DO THIS! If you do, you're an ass-hat.

Besides being unfair and a dick move, it is also stupid and even worse yet in my book - it's illogical.

What inevitably happens is that the engineers, who all have different deals, get drunk one night. Then the shit hits the fan when they tell each other what they all make and what equity they got. Come Monday morning, every engineer's password is "my_founder_is_a_dick," several viruses and backdoors are suddenly installed into the code base, and the founder gets the silent treatment - none the wiser of his impending doom. Way to go ass-hat!

If you can't tell, this one pushes my buttons. I don't give a frog's fat ass how good a negotiator an engineer is when I'm interviewing them. I want them to have such pristine code that it makes my other engineers cry. I want them to have a beautiful mind that can use logic

and reason to solve the engineering challenges that I hand them. It is completely irrelevant how good they are at negotiating.

Create a Dominant Market Share - one of the things that I took notice of was Google's move to develop a lot of their tools in Python.

Curious, I thought. Why would they do that? At the time Python was a new(ish) language, although growing quickly in popularity. Then it hit me. They were going for a dominant market share in a specific talent pool. If you can get in on a new talent pool trend, the benefits can come back ten-fold.

Here's the strategy. Get the first luminaries in the field, then as that language grows in popularity you are labeled as the de facto place to go if you want to code in that language. Then hiring get 10 times easier.

Brilliant.

In 2005, when we founded Powerset, we realized Ruby was the new Python, so we went after some A-level people in the Ruby community. The top two we went after were first, Kevin Clark (a 20



year-old wiz-kid who we were trying to convince to quit school) and second Tom Preston Werner (now the founder of GitHub).

We got both of them, and within a matter of months, we had one of the largest Ruby teams on the planet.

Anyone who wanted to code in Ruby knew about Powerset simply from the Ruby meetups which were dominated by either Powerset or Twitter people.

We then did the same thing in the field of computational linguistics. At one point we estimated that of the 200 or so people that really understood computational linguistics in the world, we had about 40 of them.

What's the benefit? Once we knew we had this level of talent market share penetration, we had almost a guaranteed worst case scenario that most startups would dream about. We knew that our talent pool was so strong, that even in the event that we just ran out of money, one of the big three search engines would simply buy us for our team.

At that time we knew that a talented engineer in a tough to get tech was worth about \$1.5 million per head. Thus, I knew with relative assurance

that since we were going to hire at least 70 people with our Series A money, that our worst case scenario was about a \$100 million exit. If anyone is paying attention, you are now saying, wait a minute! Didn't Powerset sell for \$100 million to MSFT? Yup, we nailed our worse case scenario!

Why Buy Silver, when you can Rent Gold - from realizing it was the only way I was going to hire people at Powerset.

When I am hiring A-level people, I go into the process knowing that I won't be keeping them forever. In fact, whenever I ask an engineer where they see themselves in 5 years, I often hear them say that they want to start their own company.

Rather than avoiding this during the interview process, I recommend you hit it head on and embrace it. Talk about it openly and honestly during the interview process and you'll find that not only will you increase the respect the candidate has for you by having the balls to bring it up, but you'll actually increase your chances of getting that person to join you.

I can't tell you how many people I interviewed at Powerset who told

me they really didn't want to join because they wanted to startup their own company. Whenever anyone told me that, I would go in for the kill and not let them leave until they agreed to join. My pitch to them was simple. "Let me rent you for 6 months. You'll learn an insane amount about building a startup from me that will help you prepare to start up your first company. You'll have the pedigree of Powerset standing behind you; and you'll have some cash in your pocket."

This worked incredibly well, and when those people went to create their own companies (including Serious Business, GitHub, Crowdfunder, InfoAxe and others), they were that much more prepared. I found that by allowing the superstar to know that I was willing to not only let them go, but I was willing to help them go, it made a big difference.

I don't think we could have created our team without this one principle working.

Be Transparent - me.

How to convince a superstar to join you:

"Let me rent you for 6 months. You'll learn an insane amount about building a startup from me that will help you prepare to start up your first company. You'll have the pedigree of Powerset standing behind you; and you'll have some cash in your pocket."

I have always been ultra-transparent and I have never been bitten by it. I really don't see why founders aren't more transparent about simple things, like what their founders' equity ownership is, and what their salary is in the company.

I often go much further than this and share the entire salary plan and equity plan for all employees with everyone in the company. It does something incredibly important. It will convince engineers, who are generally mistrusting, conservative and logical people, that you thought through the salary plan and equity logically and the fact that you shared it with everyone will help them trust you.

Whenever I do stuff like this, it gets a lot of attention, but I really don't understand it, I just think this is normal and a no-brainer. It also heads off at the pass that drunken disclosure scenario that always gets someone pissed off or quietly erased.

Try New Shit All the Time - trial and error.

I have found that there is some sort of weird assumption that certain things can't be changed once a decision is made. I guess it has to do with looking bad, or some weird reverence to not trying something new.

I've learned over the years to challenge that type of thinking when I come across it. For example, I have had several, but not many, cases where a specific employee was so talented that I was constantly unable to put them in a competency level, which is what we used to determine salary and equity. They were such fast learners that within months they had gone from a junior employee in my mind to an "I can't live without them" employee.

What do you do?

The answer is simple. Pay them more and let everyone know that you are doing so. Don't shy away from it. If you discover a rock star, pay them. Your employees, your board and your conscience will support you.

Further, if you put someone too high and you realize your mistake, tell them and move them down. I've done that too and the employees, including the one I moved down, agreed with me and applauded my bravery for admitting that I screwed up.

Founders Don't Interview Candidates, it's the Other Way Around - me, after an interview with Chad Walters lasted 8 months (I'm not kidding).

As a founder, you need to come to grips that killer engineers are people who are not easily convinced. Especially of you. As I mentioned before, put your team in control over who gets invited to join the cult. If you do, then you'll realize that by the time the person gets to you, they know and you know that you want to hire them. If that's true, then you'll find the candidate will want to interview you.

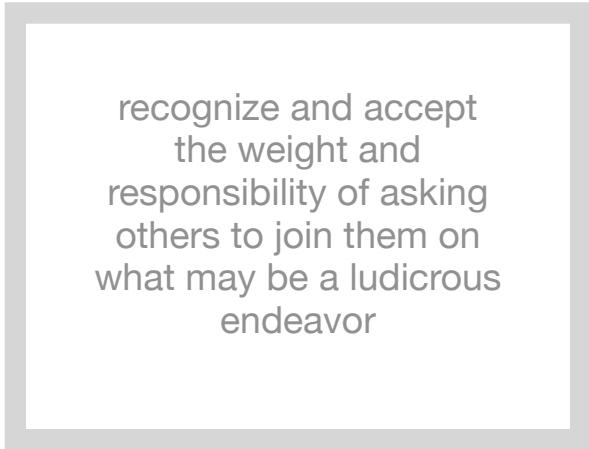
Let them.

Remember, you are a founder, which takes a special kind of risk-blindness crazy - it's what is different between the DNA of a founder and the DNA of employee #1 and onward.

Killer engineers are going to be cautious, and rightfully so. Go into the interview and remember that this person has an entirely different risk profile than you, that they will have other options if they are truly talented and that they are looking to see if you are the real deal, not a poser. For you, you have already made the decision to risk everything to do your idea; for them, this is your idea and you are asking them to risk their career for you.

So many founders get so caught up in the frenzy of creating their dream that they fail to take a moment to recognize and accept the weight and responsibility of asking others to join them on what may be a ludicrous endeavor. Be the one that sees the situation through the eyes of the person you are hiring. Listen to

them, to their concerns. Don't be a cocky dick to them, it's not the right time or the right thing to do. In this situation, you are subservient to them, you are there to answer their questions. Show them you have really thought through their concerns, that you have used rational thinking and logic. Show



recognize and accept the weight and responsibility of asking others to join them on what may be a ludicrous endeavor

them that you have more than just a great speech and a good TechCrunch article to back your claims up (most really good engineers have their prove-it-to-me-radar on full-tilt if you hype

yourself up too much - a lesson I'm sure my co-founder at Powerset has learned). Show them you have a plan that is well-thought-through, and does not rely on luck, or hope. Show them you are the real deal or they will walk.

Never Let Go - *If you find a person you want to join your team, then never let go.*

It doesn't matter what it takes. Those that know me, or have been at the 3rd Street Grill in Soma (where I do a lot of my hiring), know that I



am an animal when it comes to hiring. I don't beat around the bush, I tell them that I want them to join and that I am willing to do whatever it takes to get them.

I have been interviewed by candidates' wives, husbands, girlfriends and even moms in order to go over all of their concerns. I have brought in a shaman monk to bless a desk and even tried to get someone that red swingline stapler from "Office Space" to get them to join.

I just don't let people go away. In the entire time I was trying to hire people at Powerset I can only remember losing two guys once they got to the final round with me.

When you are building your team, be relentless, be a hiring monster and never let go of someone once you know you want them.

Role of the Press -
I'm still figuring this one out.

I can't argue with the fact that a good TechCrunch article can help attract a lot of

applicants, but I'm not convinced that it is anything other than filling your own affirmation cup. When Powerset first announced itself, I admit that I was very happy to see how much VB and TC hyped us up. It got us a lot of attention and a lot of people applying to the company - plus it felt really good.

After our first TC article we were getting almost 1,000 people a week applying. However, as time went on, I found myself hating the press we were getting, and we were getting a lot. It ended up really just screwing up my interviewing process. Engineers are by nature skeptical people. All of the press hype actually only made convincing them harder to do... there's some sort of weird anti-correlation between press hype and ease of

convincing an A-level engineer that your technology is real.

I do have to admit I did support our hyped up press strategy in the beginning, but after a year of it, it ended up being a

huge pain in the ass. It's well known that my co-founder Barney loved the press and loved the hype

press can help with hiring, but don't fall into the trap of need it's affirmation. Needing press to feel good is the hallmark of the second- hander



and to this day argues that his hype strategy was valid. Ultimately it was Barney's interpretation of how to use the press for hype, and the reversal of my opinion on this role that caused a huge disagreement between us.

I think there is just too much hype in the world. The press, who needs to get page views to make money, is forced to sensationalize their stories. It's easy for founders to take advantage of this and too often I see founders fall into the trap of going after glory, fame and power rather than going after what they should - which is simply building something awesome.

While I think that TC and VB can be used to attract talent, be wary of this tactic as it can often be the crutch of the second-hander. (re. Ayn Rand)

Practices

The following items outline some of the innovative ways in which I have implemented the principles above in my companies. Many of these were created at Powerset and I want to give special credit to Ryan Ferrier, who was my right hand man in refining and implementing many of

these practices. He was crucial in building the team.

Try Before You Buy (TBYB) -
(principles: interviews don't work and hire slowly and fire quickly)

After building so many companies, I now advise that, for as many people as possible, candidates are never hired after being interviewed.

I urge founders to offer people a one-month, auto-terminating, consulting contract that pays the person the exact amount of cash and equity (but in NSOs instead of ISOs) that the person would get if they were an employee.

During the month, the team, including people that didn't get to interview with them, would be able to interact with that person. After a full month, everyone in the company would at least have a strong opinion on whether or not the person was a cultural fit and the team that directly interacted with the person would have a strong opinion on the competency of the person.

At the end of the month, I had three options:

1. Let the contract expire without giving them an employment offer.



Notice, in this case I set myself and the candidate up so that the default was they were terminated, so the candidate who doesn't get an offer doesn't have to go through any embarrassing situation - they could simply claim they let the contract expire, minimizing the impact on the team and them. In reality you are firing this person, but this way there are no worries of getting into a lawsuit for wrongful termination or anything like that - really only employees can sue for that kind of stuff, and believe me... it is a real risk when you are firing someone these days.

2. I may simply offer to hire the person and then offer to convert their NSOs earned as a signing bonus of ISO or RSPA 83b'd stock.
3. I may make adjustments to their offer. It is not uncommon to love a candidate, think they are competent, but either under-assess or over-assess their skill level. In many cases, I have seen a person come into their try-before-you-buy contract assessed at a Junior 1 level, but after one month, their team tells me the person is really a Junior 3. Only

one time have I had it go the other way... but in all cases the candidate accepted the changed assessment.

TBYB is perhaps one of my most powerful tools. It lets you slow down the hiring process, speed up the firing process (or rather the termination process) and it let's you use interviews not as a hiring decision, but rather as one gate in a multi-gating process of a candidate getting an offer letter.

Founder's Veto - (principle: team ownership)

When building a team, the pathway to TBYB and ultimately employment should be simple. Up until about 50 employees, all employees must be voted into both TBYB and employment without any vetos.

1. By giving everyone who interviews before and during TBYB the same founder's right to veto, you are empowering them with a sense of ownership in the company. What you will find is that your A-level employees will be twice as hard to convince as you for new employees and when someone does get voted in, the level of excitement is high and

stays high for that individual.

2. Giving the founder's veto right will not only ensure the competence of each employee, it will enable you to protect your culture with an iron fist. In fact, of the times I've personally implemented TBYP, the most common reason why people don't make it through is actually not competence, it's culture. Weeding out people who are not a cultural fit is equally important to maintaining A-level competency.
3. It protects against founders making stupid decisions. One time, not me, but another founder in one of my companies, thought it would be a good idea for one of his direct family members to interview. Now that's fine in most cases, but what was interesting was that my personal opinion was that this person was nowhere near qualified, but I let them go into the process anyway.

founders' veto rights enable employees to protect, while the double unanimous vote empower employees to promote

I wanted to see if what I thought was a clear veto would occur. It did, but that's not why I bring it up. The person that pulled the veto was our most junior engineer at the time. It was one of the proudest moments I had during the process of building my team. In the face of the power of a founder, I had created a process that made a junior employee not only feel like they had a right to stand up against a founder, but that I actually gave them that power and they used it - ultimately to protect the team. That one moment let other junior engineers and even non-technical employees know that when it came to adding a new person to the team, they had the power, the same exact power as a founder, to protect their team. Other instances followed and I can say with certainty that the veto was never a bad decision.

Double Unanimous (principle: team ownership)



Although many confuse this with the founder's veto practice, it isn't the same.

It's the inverse.

Not getting a veto is a passive event, getting a unanimous "yes" is an active event. Thus, in order to get an employment offer a candidate must actually not only not get a veto, but also get a double unanimous "yes" vote.

The reason I recommend this is that the motivation for vetoing and the motivation for actively saying "yes" is different.

Giving veto rights is equivalent to giving protective rights to existing employees.

Making them all say "yes" gives them responsibility and ownership over the decision to add a new person to the team. If the person winds up not fitting in or not being competent, then everyone takes responsibility. Thus when mistakes happen, which they will, the entire team takes ownership and the team gets stronger as a result.

Beyond the impacts to the team, there is a dramatic impact on the candidate. Imagine what it feels like for them. They interview for a week,

to get to TBYB and then have to work for a month to get employment. At first blush it looks like a great way to not get people applying to work for you, but the effect is actually the opposite. If you were an A-level player and you knew the process involved in joining was this difficult, then you would have a high degree of confidence that the people who made it through this process were rockstars and thus you would want to work with them. If you ask most rockstar engineers what they worry about most when joining a company, the answer is consistent, they want to work with other *rockstars* on really interesting projects. B-Level engineers generally have the exact opposite reaction - complaining about how difficult the interview process is. They generally shy away from companies with strict hiring policies. Let them - you will benefit more. Now imagine what C-level people think of it.

Finally, imagine what it feels like once a person makes it through TBYB and gets an offer for employment. Double Unanimous is a right of passage and every person that gets an offer to become an employee is aware what got them there. For any person that makes it to this point, it creates an incredible



sense of empowerment and affirmation that they are the best in their class.

The last effect that it has is that it changes the dynamic between you and the candidate when it comes to the all important founder/candidate meeting. In short it makes your job much easier.

They are on top of the world because they just made it through their right of passage and passed. It is and should be one of the highlights of their career to have made it. So who is going to say no to an employment offer at that point?

The end result is that the acceptance rate to employment offers is incredibly high when using this technique. It makes your budgeting and HR spreadsheets much more predictable, thus making you look good, but more importantly it makes you look like a brilliant rockstar cult leader. The truth is that by installing this process, you ensure victory. It reduces the dependency of the cult of your individual founder cult of personality and utilizes the power of your team - the real cult of personality.

No Negotiation - (*principles: hire for talent, not negotiating skills; be transparent*)

One of the things that is incredibly important for a founder in today's engineering world is to create a logical and rational hiring plan with salaries and equities mapped out.

1. Outline 10 levels of competency per talent type; i.e. one for each type of engineer, one for designers, one for product people and so on.
2. Create your high and low salary bookends for each competency, which may be unique depending on your industry.
3. Then create 10 levels of salary for each competency: Junior 1-3, Mid 1-3, Senior 1-3, and a VP. Then all Junior 2 engineers would have identical salaries, as would all senior 3 engineers, etc.
4. For your ESOP pool, you will have a specific number of shares to allocate - usually around 20% at or just before Series A. With your VC develop a timeline of how long that pool should last. If you are planning to raise several rounds of financing, plan out how much ESOP should be left

right before each round of financing, keeping in mind, new investors typically apply pressure for the the past investors and existing employees to take the dilution needed to shore up the ESOP pool.

5. Once you have your pool targets, then work with your board to create a general plan of how many employees, and of what type, you want right before each round.

sharing your logical reasons for paying people what you pay them will foster trust

6. Then back into how much equity each competency level gets using the following methodology. Start off by using relatively standard equity disbursements for each type of role - VCs can help, they have many, many stat sheets on this one.
7. Then, after you have your beginning equity allotments per strata, take the remaining equity to be allocated in your ESOP and dole that out to each strata based on your hiring plan. Remember to reduce the amount of equity each strata gets over time by the proportion by which you believe

your total company value will grow; i.e. make it so that while your ESOP stock's estimated stock price is going up over time, you are reducing the amount of shares you are giving out.

Therefore, if you gave employee A, who is a junior 1, 10 shares at angel funding and you estimate your stock price at .01 and you are hiring employee B who is also a junior 1 at Series A, when common stock is value at 10 cents, you would give that employee one tenth the

stock. The point is, you're giving the same or less starting cash equivalent to both people because people that joined earlier took more risk and therefore should get more reward. There are many ways to do this, but it is important that you reduce the amount of stock issued over time so that earlier employees get equivalent or better value.

It is important that you do this in a systematic and rational way so that you can show it to your employees and new candidates. Once you have the spreadsheet developed, follow it as best you can. Be strict on following the salary amounts and



do your best on the equity. If you need some sort of sweeteners to get someone on board, do your best not to budge on the core compensation.

Sometimes you can hold 83b elections as a benefit, or an offer of RSP instead of ISOs, but never adjust someone's salary within a level to get them on board - the other engineers will know you did so and kill you. One of the big benefits of using this strategy to hire people is that engineering candidates on whole have a very positive reaction to hearing that there is no negotiation.

1. Engineers generally hate negotiating anyway and if there was one word I would use to describe their general reaction, it would be "relief" on the candidates part.
2. Engineers are highly logical people and if you email them your spreadsheet and explain that you negotiate with no one and that their peers in the company determine their skill sets and thus their strata, not HR, then all that's really left is for them to check your logic in your spreadsheet. If it is sound, then I have found it removes perhaps the most awkward and stressful part of the hiring process and turns it into a

logical and rational process - which engineers like.

3. Further, you will like it as well, it will help prevent you from doing stupid things that you regret and get bitten in the ass for later anyway.
4. It will also enable you to have a very different conversation when the candidate gets to you. Instead of this part of the process being confrontational, where you are the opposition, there is no opposition. You and the candidate can simply talk about what's really important. Then interviewing you.

Offer them the Peter Thiel

Deal (principle: *try new shit all the time*)

I remember at a board meeting at Powerset Peter Thiel asked us how the morale of the company was going. Like all founders at board meetings, we said it was going great.

Peter said "*let's prove it.*"

What we did is offer the following:

1. Anyone can voluntarily, permanently reduce their salary

by one or more strata levels,

2. For each competency level forfeited you get XX,XXX more ESOP shares.

By doing this we hypothesized that we could actually measure the excitement in the

company by how many people took us up on the offer. To our great relief, a lot of people did. The company benefited by getting an improved information state at a critical

moment in the company (like right before you ask them to walk into fire) and we also got to save cash flow. The employees benefited in that those that volunteered to do this got a lot of extra stock. Stock they may have either never gotten or would have had to wait years for as they moved up the ladder. We even had some employees attempt to give up their entire salary - we had to tell them that legally we had to pay them minimum wage - so that's what they did. It really felt good to have these types of problems.

using the Peter Thiel deal can be effective in determining how pumped your employees are to be in your company

Miscellaneous: 83b, Game Rooms, Food, Big Monitors, SOMA, yada, yada

Everyone else talks about these things, blah blah blah, so I won't bother except to say this: you should know it by now. If you don't, then

Google it (note: I'm also contractually obligated to tell you to Bing it as well- just kidding).

One Mile Rule - (principles: try new shit all of the time)

While at Powerset, we had heard Facebook was paying people more if they lived within one mile of the office. So we did it too, just to see what would happen.

All I can say is WOW!

We paid people on average about \$1,500 more per month. What happened was the people that moved:

1. No longer had a commute, so each person captured hours back for their life



2. They were happier because they had less road rage and less drama in their lives
3. The spouses, girlfriends, boyfriends and pets were much happier
4. They stayed longer at the office because they knew they could always just run home
5. They were much more productive
6. Once some people did it, other copied and
7. We got way more value each month than the \$1,500 per person we paid.

The way it made the team gel, the productivity gains and just the sense that everyone was kind of living the same life mattered. It had a much larger impact that I initially thought. You could implement this anyway you want, but I always recommend it. If you don't have money, offer equity, hell offer bigger monitors. Make the benefit real, but make it attractive for everyone to live nearby.

Know Your Worst Case Scenario -
(principle: you are being interviewed)

By the time any candidate gets to you, they are going to have lots of questions. One of them is going to be "what's the worst case scenario?"

DO NOT MESS THIS UP!

You need to demonstrate that you have taken a realistic, rational and logical look at this case. For many engineers at Powerset, they would actually make me walk them through blow by blow what the worst case would look like, and to describe the financial implications, reputation implications and technology usage implications as it related to them. On many occasions this even involved building a spreadsheet with them and walking through my logic, it sometimes even involved providing proof of things such as "each engineering head was worth \$1 to \$2 million."

Many of the engineers at Powerset still probably remember my worst case with them, because I often used it as my primary selling point when I went to hire them. I built my worst case scenario based on simple and logical truths that had been constructed purposely from the very outset of the company. Here it is: "the worst case scenario of Powerset is better than the middle

case, and in some cases the best case, scenario of the other startups that are trying to hire you. Because:

1. We have a killer team, that team has intrinsic value no matter what, and the network of relationships you will build in this company will last you a lifetime
2. You will enjoy working with our team because we only hire A-level engineers like you, and you know how hard it is to get a gig here
3. The technology we are building is novel: we are building a search engine from scratch and the problems we face will challenge your engineering skills more than any company that is trying to hire you.
4. If you join us, you will have a very real impact on something important; we are not some web 2.0 Rackojoo company flipping bits for ad sales.

Showing a risk-averse candidate that you have actually thought about ensuring a good outcome - even in the worst case - can be the thing that convinces them to join

5. The worst case scenario financially is that we simply run out of money (which by the way will take 2 years at least based on what we have already raised) so the worst case is that for two years you get all of the above and then we sell for about \$100 million which would mean every person in the company could put a down payment on a new house
6. From a reputation standpoint, the worst case is that you are here for two years, getting a full salary and then you will a) have the pedigree of being part of a well-known startup and b) you will be one of the most sought after engineers on the planet because you will have had experience in either Search, Ruby or Computational Linguistics which are going to be skills nearly every company in the world will be seeking in two years or less.

If you analyze my worst case scenario, you'll notice that in most of it I simply restate what the



candidate already knew in their mind. Really the only new piece of information in the whole statement was the estimate for how much we would go for if we sold the company and that was simply based on an estimate that any engineer could derive simply by doing a Google search on the value per engineering head in an acquisition.

In short, I made sure that while my statements were bold, and perhaps may seem cocky, they really weren't. They were logical and rational conclusions based on simple and well-backed assertions. I will have to say that not all engineers agreed with me, but every single person appreciated that I didn't say something stupid like "trust me.... it's going to be awesome.... I swear." No matter what your response is to this question, you need to be able to nail it. While you may be less inclined to be as bold as I was in my statement, my point is separate. Take the time to come up with an answer you believe in. I believed in mine and I think it showed through when people talked to me, and that made all of the difference in the world.

4:20 - (*principle: transparency*)

It's called many things in many of the companies I have influenced. Powerset called it "4:20" or "Beer: 20;" Virgance, Carrotmob and 1BOG call it "naked lunch;" Serious Business called it "beer Fridays;" and I think Crowdfunder calls it "demo Fridays".

At Powerset, where I started it, it went something like this. We initially tried to have it at 4pm, but for some reason the Data Center team and the Tools team needed 20 minutes to "prepare" uhm hmmm... then once everyone was "prepared" we would generally herd them into the game room and give them some beers. Chris Van Pelt, now founder at Crowdfunder, would often start the meeting off with a magic show. Yes, a real magic show. Generally he would try to involve some small explosion or two. Then with the crowd warmed up, one of our employees would demo something awesome that they worked on that week - sometimes it was technical, sometimes it was just talking about whether or not we were looking at an east coast "Tornado" style foosball table or a west coast table.

Once the demos were done and the the applause stopped, I would sit



down on what was called the hot seat. It was a sort of red stool thing.

During this time, there were only two rules:

1. Anyone could ask any question.
2. I had to answer.

And let me tell you, they would unleash on me. Some of the questions were stupid at first, like "when was the last time you cried?" or "how much money is in your bank account?" or even "when did you lose your virginity?" to which my responses were 1) I had to admit that sometime a really good dish of heuvos rancheros made me tear up, 2) I didn't know because my wife takes care of that stuff and 3) HR VIOLATION!...

But after the initial silliness was done, we would really get into it and everyone from the most senior engineer to the most junior office manager piped in. The point of 4:20 was that it was a time for us to share our successes through our demos and to maintain and sometimes strengthen the bond between me and everyone else in the company.

In order to succeed, you need to build an incredibly tight unit. There will be times when you will need to

ask each and every person on your team to make sacrifices, whether the sacrifice be time at work as opposed to time at home, or helping another engineer whose code blew up the data center and accidentally launched a Russian nuke into orbit or even asking your team to build something over from scratch because you, late last night, decided to completely change direction.

The degree to which your team is willing to respond to your call affects your ability to succeed perhaps more than any other factor in a startup. If these folks are going to follow you, then you must serve them well. The best way to do that is to understand that while you are their leader and you will ultimately make all of the tough choices, you are not above questioning. In fact, you should go into it with the perspective that you deserve and demand to be questioned.

It will make you stronger because you will know that you have to prepare. It will make you team stronger too, because it will give them a real sense that the company is theirs as much as it is yours. Lastly, it will make the whole company stronger as a unit, as a cult, because you will no longer have any sense of "management," of "them" or of "they" to blame. Your



cult's battle cry should embody the sense that you and your team are one and that your success or failure is completely owned by all.

Your battle cry: "We are They. They are We. And We Will Conquer."

(note: you may also shout "We Are Sparta!" during any battle cry, as it is always considered appropriate to do so).

Conclusion

There are many things that can determine the success or the failure of a startup, but no factor is more powerful and more in your control than you and how you build your team. Regardless of the type of company you are building, the strength, agility and cult-ness of your company should always be a top priority.

As you are creating your company, check in with yourself every month to see if you are following the principles and the practices that will lead to building your own cult. The minute you find a weakness, get rid of it. The minute you realize you have made a mistake, admit it and then fix it. The minute you feel out of touch with your team, dig your heels back into the ground and get back on track.

If you can do these things, then your chances of executing well increases not just a little bit, but exponentially.

Above all, realize that if you do create a strong cult, it will make you a better leader as well. The stronger your cult, the more it will challenge you; the more your people will demand that their leader be worthy of leading them. You, as a founder and leader, must be willing to make whatever sacrifices are necessary to deserve to be your own team's leader. Remember, it is not your birthright to lead your team just because you founded your company. You must earn it every single day.

If you built a cult and earn your keep then you will achieve something more powerful than the title of CEO, more important than the wealth gained from a successful exit.

You will become the leader of your own cult.

(note: if you would like to comment on this article or any other article feel free to do so on the Blognewcomb Facebook Page wall by clicking [here](#)).