

[pmarchive.com /guide_to_startups_part4.html](https://pmarchive.com/guide_to_startups_part4.html)

The only thing that matters

12-15 minutes

Jun 25, 2007

This post is all about the only thing that matters for a new startup.

But first, some theory:

If you look at a broad cross-section of startups -- say, 30 or 40 or more; enough to screen out the pure flukes and look for patterns -- two obvious facts will jump out at you.

First obvious fact: there is an incredibly wide divergence of success -- some of those startups are insanely successful, some highly successful, many somewhat successful, and quite a few of course outright fail.

Second obvious fact: there is an incredibly wide divergence of caliber and quality for the three core elements of each startup -- *team*, *product*, and *market*.

At any given startup, the team will range from outstanding to remarkably flawed; the product will range from a masterpiece of engineering to barely functional; and the market will range from booming to comatose.

And so you start to wonder -- what correlates the most to success -- *team*, *product*, or *market*? Or, more bluntly, **what causes success?** And, for those of us who are students of startup failure -- **what's most dangerous: a bad team, a weak product, or a poor market?**

Let's start by defining terms.

The caliber of a startup *team* can be defined as the suitability of the CEO, senior staff, engineers, and other key staff relative to the opportunity in front of them.

You look at a startup and ask, will this team be able to optimally execute against their opportunity? I focus on effectiveness as opposed to experience, since the history of the tech industry is full of highly successful startups that were staffed primarily by people who had never "done it before".

The quality of a startup's *product* can be defined as how impressive the product is to one customer or user who actually uses it: How easy is the product to use? How feature rich is it? How fast is it? How extensible is it? How polished is it? How many (or rather, how few) bugs does it have?

The size of a startup's *market* is the the number, and growth rate, of those customers or users for that product.

(Let's assume for this discussion that you can make money at scale -- that the cost of acquiring a customer isn't higher than the revenue that customer will generate.)

Some people have been objecting to my classification as follows: "How great can a product be if nobody wants it?" In other words, isn't the quality of a product defined by how appealing it is to lots of customers?

No. Product quality and market size are completely different.

Here's the classic scenario: the world's best software application for an operating system nobody runs. Just ask any software developer targeting the market for BeOS, Amiga, OS/2, or NeXT applications what the difference is between great product and big market.

So:

If you ask entrepreneurs or VCs which of *team*, *product*, or *market* is most important, many will say *team*. This is the obvious answer, in part because in the beginning of a startup, you know a lot more about the team than you do the product, which hasn't been built yet, or the market, which hasn't been explored yet.

Plus, we've all been raised on slogans like "people are our most important asset" -- at least in the US, pro-people sentiments permeate our culture, ranging from high school self-esteem programs to the Declaration of Independence's inalienable rights to life, liberty, and the pursuit of happiness -- so the answer that team is the most important *feels* right.

And who wants to take the position that people don't matter?

On the other hand, if you ask engineers, many will say *product*. This is a product business, startups invent products, customers buy and use the products. Apple and Google are the best companies in the industry today because they build the best products. Without the product there is no company. Just try having a great team and no product, or a great market and no product. What's wrong with you? Now let me get back to work on the product.

Personally, I'll take the third position -- I'll assert that *market* is the most important factor in a startup's success or failure.

Why?

In a great market -- a market with lots of real potential customers -- the market *pulls* product out of the startup.

The market needs to be fulfilled and the market *will* be fulfilled, by the first viable product that comes along.

The product doesn't need to be great; it just has to basically work. And, the market doesn't care how good the team is, as long as the team can produce that viable product.

In short, customers are knocking down your door to get the product; the main goal is to actually answer the phone and respond to all the emails from people who want to buy.

And when you have a great market, the team is remarkably easy to upgrade on the fly.

This is the story of search keyword advertising, and Internet auctions, and TCP/IP routers.

Conversely, in a terrible market, you can have the best product in the world and an absolutely killer team, and it doesn't matter -- *you're going to fail*.

You'll break your pick for years trying to find customers who don't exist for your marvelous product, and your wonderful team will eventually get demoralized and quit, and your startup will die.

This is the story of videoconferencing, and workflow software, and micropayments.

In honor of Andy Rachleff, formerly of Benchmark Capital, who crystallized this formulation for me, let me present *Rachleff's Law of Startup Success*:

The #1 company-killer is lack of market.

Andy puts it this way:

- When a great team meets a lousy market, market wins.
- When a lousy team meets a great market, market wins.
- When a great team meets a great market, something special happens.

You can obviously screw up a great market -- and that has been done, and not infrequently -- but assuming the team is baseline competent and the product is fundamentally acceptable, a great market will tend to equal success and a poor market will tend to equal failure. **Market matters most.**

And neither a stellar team nor a fantastic product will redeem a bad market.

OK, so what?

Well, first question: Since team is the thing you have the most control over at the start, and everyone wants to have a great team, what does a great team actually get you?

Hopefully a great team gets you at least an OK product, and ideally a great product.

However, I can name you a bunch of examples of great teams that totally screwed up their products. Great products are really, really hard to build.

Hopefully a great team also gets you a great market -- but I can also name you lots of examples of great teams that executed brilliantly against terrible markets and failed. *Markets that don't exist don't care how smart you are.*

In my experience, the most frequent case of great team paired with bad product and/or terrible market is the second- or third-time entrepreneur whose first company was a huge success. People get cocky, and slip up. There is one high-profile, highly successful software entrepreneur right now who is burning through something like \$80 million in venture funding in his latest startup and has practically nothing to show for it except for some great press clippings and a couple of beta customers -- because there is virtually no market for what he is building.

Conversely, I can name you any number of weak teams whose startups were highly successful due to explosively large markets for what they were doing.

Finally, to quote Tim Shephard: "A great team is a team that will always beat a mediocre team, given the same market and product."

Second question: Can't great products sometimes create huge new markets?

Absolutely.

This is a best case scenario, though.

VMWare is the most recent company to have done it -- VMWare's product was so profoundly transformative out of the gate that it catalyzed a whole new movement toward operating system virtualization, which turns out to be a monster market.

And of course, in this scenario, it also doesn't really matter how good your team is, as long as the team is good enough to develop the product to the baseline level of quality the market requires and get it fundamentally to market.

Understand I'm not saying that you should shoot low in terms of quality of team, or that VMWare's team was not incredibly strong -- it was, and is. I'm saying, bring a product as transformative as VMWare's to market and you're going to succeed, full stop.

Short of that, I wouldn't count on your product creating a new market from scratch.

Third question: as a startup founder, what should I do about all this?

Let's introduce *Rachleff's Corollary of Startup Success*:

The only thing that matters is getting to product/market fit.

Product/market fit means being in a good market with a product that can satisfy that market.

You can always feel when product/market fit isn't happening. The customers aren't quite getting value out of the product, word of mouth isn't spreading, usage isn't growing that fast, press reviews are kind of "blah", the sales cycle takes too long, and lots of deals never close.

And you can always feel product/market fit when it's happening. The customers are buying the product just as fast as you can make it -- or usage is growing just as fast as you can add more servers. Money from customers is piling up in your company checking account. You're hiring sales and customer support staff as fast as you can. Reporters are calling because they've heard about your hot new thing and they want to talk to you about it. You start getting entrepreneur of the year awards from Harvard Business School. Investment bankers are staking out your house. You could eat free for a year at Buck's.

Lots of startups fail before product/market fit ever happens.

My contention, in fact, is that they fail *because* they never get to product/market fit.

Carried a step further, I believe that the life of any startup can be divided into two parts: *before product/market fit* (call this "BPMF") and *after product/market fit* ("APMF").

When you are BPMF, focus obsessively on getting to product/market fit.

Do whatever is required to get to product/market fit. Including changing out people, rewriting your product, moving into a different market, telling customers no when you don't want to, telling customers yes when you don't want to, raising that fourth round of highly dilutive venture capital -- whatever is required.

When you get right down to it, you can ignore almost everything else.

I'm not suggesting that you *do* ignore everything else -- just that judging from what I've seen in successful startups, you *can*.

Whenever you see a successful startup, you see one that has reached product/market fit -- and usually along the way screwed up all kinds of other things, from channel model to pipeline development strategy to marketing plan to press relations to compensation policies to the CEO sleeping with the venture capitalist. And the startup is still successful.

Conversely, you see a surprising number of **really well-run startups** that have all aspects of operations completely buttoned down, HR policies in place, great sales model, thoroughly thought-through marketing plan, great interview processes, outstanding catered food, 30" monitors for all the programmers, top tier VCs on the board -- **heading straight off a cliff due to not ever finding product/market fit.**

Ironically, once a startup is successful, and you ask the founders what made it successful, they will usually cite all kinds of things that had nothing to do with it. People are terrible at understanding causation. But in almost every case, the cause was actually product/market fit.

Because, really, what else could it possibly be?

[Editorial note: this post obviously raises way more questions than it answers. How exactly do you go about getting to product/market fit if you don't hit it right out of the gate? How do you evaluate markets for size and quality, especially before they're fully formed? What actually makes a product "fit" a market? What role does timing play? How do you know when to change strategy and go after a different market or build a different product? When do you need to change out some or all of your team? And why can't you count on on a great team to build the right product and find the right market? All these topics will be discussed in future posts in this series.]