

[hackernoon.COM /the-build-order-every-startup-should-follow-to-become-successful-635e7ed00fa3](https://hackernoon.com/the-build-order-every-startup-should-follow-to-become-successful-635e7ed00fa3)

# The Build Order Every Startup Should Follow To Become Successful

Wayne Chang @waynechang

11-14 minutes

---

The vast majority of startups follow a very similar process.

Step 1: Come up with an idea.

Step 2: Create a prototype.

Step 3: Raise capital.

Step 4: Launch.

**In the simplest form, we could call this a startup's "build order."**

In competitive gaming (think games like Starcraft), your build order is the process in which you manage resources — the goal being to build (and compound your efforts) faster than your opponent.

Entrepreneurship is no different.

**The startup game is all about who can manage their resources more effectively.**

Which means, given all the same pieces, every startup should end up with the same perfect final product. After all, every startup needs to hire, every startup needs to design and develop their product or service, every startup needs a marketing department, etc.

**In theory, the results should be identical.**

But they're not — and they never are. Your build order changes your outcome dramatically.

For example: if you do design before engineering, that will affect the outcome.

Engineering before design affects the outcome.

Marketing before product development affects the outcome.

Why?

**Because the secret sauce of every great, successful company is its ability to take advantage of TIME.**

Time is the one variable every single startup's build order is weighed against. Time is always moving in the background — which means the name of the game is to get "time" to work for you (not against you).

Part of what has made Warren Buffett such a successful investor is his perspective on time, saying, "The 8th wonder of the world is compounding interest."

I would add to that and say there is an equally magical wonder in the startup world, and it's what I would call The Right Build Order.

**Build order, and the way you sequentially pursue your end goal, affects the outcome of the startup tremendously.**

A lot of startup founders, after they raise a big chunk of capital, think to themselves, “OK, now we need to build a team.”

And that’s great. I believe founders don’t build great products — they build great teams that build great products. That’s a fundamental truth for great startups.

However, the way to execute that varies across the board.

### **Here’s an example of a poorly designed build order for building a team:**

Founder says, “We need to hire the next engineer.”

The founder then proceeds to put together a job description and make a post on Craigslist. And when resumes start trickling in, he or she sits there reading through each one, trying to find the perfect engineer.

In this “build order” scenario, the founder has elected himself or herself as the one to sift through the resumes, make screening calls, and interview all the applicants — when, in reality, that’s a very poor use of their time. They should be focused on more strategic things, not doing something that could easily be hired for.

This might seem inconsequential, but compound the amount of time that one task takes over the course of a day, a week, a month, and you’ll start to see how inefficient it is — especially when compared side by side to a competitor with a better build order.

### **Here’s a more effective build order:**

In recognizing the inefficiencies of a founder sifting through resumes, one of the first hires of any startup that’s venture backed should be a **recruiter**.

This is a no-brainer. And here’s why.

In hiring a recruiter first, you’re essentially saying, “I’m going to trade money now for a resource that will save me time later. They are going to be our talent scout, looking everywhere for the best talent — so that I, as a founder, can get more of my own time back and work on the next piece of my startup.”

Doesn’t that sound like a much better use of resources?

To make this even more obvious, imagine you’re playing a resource-driven video game and you have two ways you can spend your first 50,000 imaginary coins. You can either spend some of those coins to buy a 2nd character that will work on finding more resources for you, or you can keep those coins to yourself and do the slow and steady building yourself.

I’d imagine any gamer would absolutely choose the former. But this is something that’s very non-obvious to a lot of founders. It’s as if they feel compelled to do as much of the work themselves.

Meanwhile, the ones that see a much bigger picture, are willing to give up some of their resources now to build a more powerful resource-generating team faster.

### **This was exactly our build order when we were building Crashlytics in 2012.**

From the onset, my cofounder [Jeff Seibert](#) and I made sure that one of our very first hires that we made was a recruiter.

And by the time we were 4 months old, we already had the team profile of a 2-year-old startup. It was one of our key advantages to being able to build so much in such a small amount of time — and then getting acquired by Twitter in its largest acquisition ever at the time just 11 months after our product launched.

Most founders don’t think about how they are hiring. They deliberately choose (within their build order) to bring in a recruiter later. But by making this one of the first items on our build order, the results resembled that of compounding interest: great teams collaborating together. And because of the calendar effect (the variable of time clocking in the background) we were able to save costs and benefit from the results of having talented team members on deck, faster.

Now, all of the above relates to the build order of a great team.

However, everything you do as a startup has a build order.

## Another, more specific build order might be, “I want to build the best product possible.”

There are products that are engineered first, and designed later.

There are products designed first, and then engineered.

These are all conscious “build order” decisions, each *dramatically* affecting the outcome.

I’ve been building products for a long time, so my perspective on a successful build order here is that you need to get the designer in the room long before a single line of code is written.

Now, this doesn’t mean that you first need the best photoshop artist, or you first need to spend a ton of money on an agency to do brilliant mock-ups of your product. I mean grabbing a pencil and paper and sketching out the wireframes, thinking through the experience.

Not how it looks. How it *works*.

We’ve all seen products that are engineered first and then later, much much later, designed is added on. The results tend to be underwhelming — and by the time the team realizes the mistake in their build order, it’s already too late. The engineer is the one leading how the product feels, works, etc., which means the designer is, to use a metaphor, “putting lipstick on a pig.”

In the startup world, there is this general belief that in order to start building something great you need pixel-perfect mockups, or you have to have deep knowledge of UI design principles, etc.

But none of those things matter — at least, not right away. In fact, I’d suggest not even opening your laptop. Just get a pencil and paper. This was an exercise I went through with a founder last week, actually. For 3 days straight, all we did was work with pencil and paper, unlocking how the application is supposed to work.

*When do I ask them to put in their information? When do I show them the analytics dashboard? Is this the right time to ask them for their credit card info?*

## Product design is 90% how it works, and 10% how it looks.

Which means, as a founder, and as a product lead, your time (and resources) should be spent focusing on how it *works*.

And the best part about this product design build order is that, once you’ve got those key wireframes drawn out, engineering can begin working on making it work, while design can start filling in pixels and bringing that functionality to life.

Now, you have two tracks moving in parallel.

That’s a great build order hack, because you’ve unlocked two resources moving toward the same goal at once — doing things in parallel instead of sequentially.

## Moving forward from there, here’s what an effective marketing build order looks like:

The 1989 classic movie *Field of Dreams* says, “If you build it, they will come.”

Unfortunately, this quote only holds true for the dreamy startups that never survive in the real world.

Startups and product designers that start their build order with this belief in mind quickly find themselves saying, “We built it. No one is here. Now we have to figure out how to market this.”

Meanwhile, calendar time is rolling, burn rate is increasing, resources are being depleted and there is no turning back.

A better build order would be to think about distribution and virality before you even build the product — so your marketing becomes part of the product itself.

In my last article, [The Quintessential Guide For Building An Unforgettable First-time User Experience](#), I talked about how to get User #2 on board and using your product, and actually building that onboarding process into the product experience.

When we were building Crashlytics, our build order integrated marketing thinking into our product's development. We thought about it way ahead of time, and wasn't just outsourced to a marketing head or an agency. It was an innate part of our product — and ended up paying dividends down the road.

You can read the full story in the article above, but in short, we let our first users introduce our second users — and so on and so forth. When we did this with Crashlytics, marketing no longer felt like this weird outside force. It was something felt internally through every team, with all of us working toward the same goal.

### **Every single item on a startup's build order has a build order within it.**

Once you understand this way of thinking, you start to realize that the process of building a successful startup happens on both macro and micro levels.

There is a build order for the company as a whole, and there are build orders for things as small as how you handle your Accounts Receivable items.

It may sound boring, but if you're a founder and you're revenue driven in any way, Accounts Receivable becomes a major thing — especially if you're bootstrapped.

This department says, "I need to charge \$1,000, then we need to perform the work." And the work, of course, incurs costs such as overhead, insurance, employee payroll, etc. And when it comes time for the client to pay, they are late — or they might not even pay at all.

As a founder, you're now stuck in a situation where you can either spend time chasing down that missing capital, or spend time working on a different project that creates new (growth) revenue for your company.

Again, this is extremely inefficient when measured against the variable of time.

A better build order here is to think in advance, "I know getting payment might be an issue, so for smaller clients, as a company we are going to require credit card payments upfront."

When they sign on, you have them agree to be charged directly. And if it's a larger engagement, you might send them an ACH form where they authorize monthly withdrawals.

If a client isn't fine with these terms, this becomes a negotiation point. But at least it is addressed at the start, instead of becoming a problem down the road.

### **As a founder, the important thing to remember is that you are always working against time.**

In every situation, you should be asking yourself, "If I do *this* before *that*, what is my outcome — and if I do *that* before *this*, can I reach the same outcome *faster*?"

There is no one single right answer when it comes to the build order of a successful startup.

The common thread between all successful startups, however, is how conscious they are of their build order process.

Don't just do what's next on your To Do list.

Be deliberate with what you choose to do *first*.