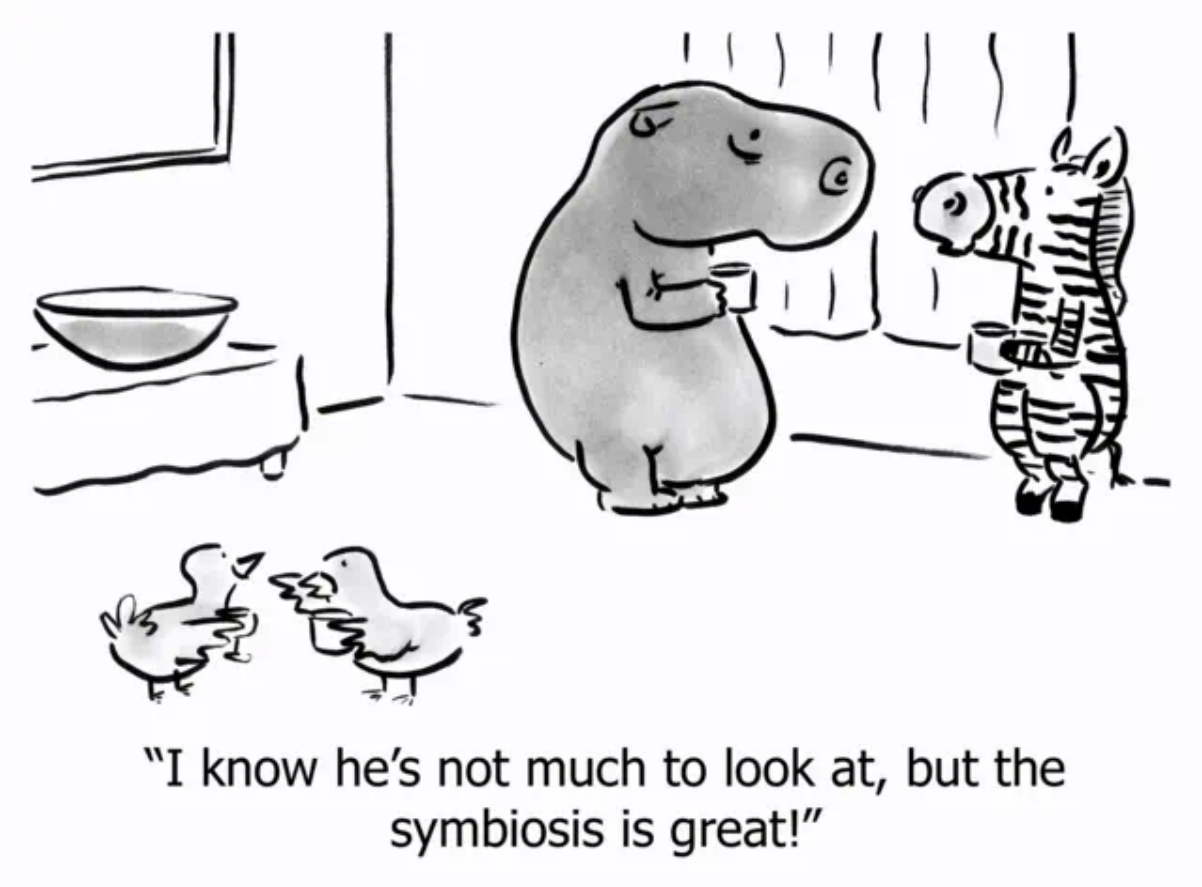
Jason Cohen
April 25, 2011
Reading time: 7 min
Printable PDF

Never say “no,” but rarely say “yes.”

“Focus” requires saying “no” to most things, but there’s a way to do it that allows you to say “yes” exactly when it matters most.



source

Everyone says small startups require focus. Say “no” to anything that distracts from your goal, your vision, your strategy, tempting though it is to explore all opportunities, hoping each time that *this* is the one that will catapult you to “ success” (whatever that is).

Lack of focus results in half-assed initiatives, each interrupted by apparently greener pastures before you’ve invested the time and devotion it deserves. Learn to say “no!”

Ah, but then again you must also experiment with new ideas. Fail fast! Pivot! Test! Doubt! Always be collecting evidence that you’re wrong, always be trying new things in case you’ve been blind. Never pass up an opportunity to change, learn, grow.

So… how are you supposed to explore other ideas if you’re also supposed to be saying “no” to anything that diverges from The Plan?

Here's what I do: I never say "no." But I carefully qualify "yes."

I learned this trick in high school. In the mid '90s it was clear that Apple had lost the personal computer battle and all their developers were fleeing like rats off a sinking ship into the ocean of opportunity that was Windows 95. As a maven of the Macintosh API and still willing to admit it, I landed small contracting jobs fixing up code that other developers wouldn't touch.

My typical rate was $25 per hour, which feels like a lot of money when you're 17 in the '90s.

One day I got a call from some poor schleps I didn't want to help. They had just completed a new product written in Java and it was broken on a Mac, and could I help? None of their customers used Macs, so they didn't think Macs were important, but then it turns out the main investor is keen on seeing the demo on a Mac, and when they tried it, it didn't work. (Yay investors! Yay supposedly-cross-platform-platforms-that-aren't-really!)

I wanted no part of this. Java was brand new and known to be full of bugs, and anyway I was a C/C++ kind of guy, and I didn't want to get involved in an academic fad language like Java[1].

[1] So yeah, I simultaneously decided that (1) Java is a fad and (2) I'm sticking with the Macintosh Toolkit; five years later one those platforms had zero developers and the other had one million, and I picked exactly wrong.

Predicting the future is hard.

I could have said "no." Given my specialty and my goals, traditional career (or startup) theory says I should have said "no."

But instead, on the advice of an older, wiser friend, I showed up at their office and said I'd do it for $100 per hour.

I fully expected them to laugh in my face. Maybe I would receive a condescending talking-to about the audacity—nay, the *impudence!*— of someone of my tender age and meager experience walking in here and demanding such outrageous compensation, someone who, let's be clear, is technically too young to even enter into a legal consulting agreement in the first place.

And then I would have slinked out of there, embarrassed but ultimately no worse for wear.

But that's not what happened. They looked me up and down, their faces painted with both incredulity and surrender. They said OK. An hour and a half later, everything was working. The difference between saying "no" and getting $150 for about two hours of my life was all in how I phrased "yes."

And I have stories that went the other way, which are just as important.

At WP Engine, for example, we're constantly talking to large bloggers who want to move to our system. These are folks with big requirements—tens of millions of monthly page-views, traffic spikes, custom code, perfect up-time, and 24/7 support.

Should we take on those clients? Maybe not—after all the stated goal of WP Engine is to serve the "middle market"—the folks who have outgrown free blogs, don't like maintaining their own servers, but aren't so large that they have extreme hosting demands. That's our profitable niche[2].

[2] Editor's note: This was written about 1 year into WP Engine's life; in 2023 WP Engine is 13 years old with 200,000 customers, and powers more large WordPress sites than anyone on Earth. This *was* a great segment to start off with, but we evolved as we scaled.

Or so we think. But if we just say "no" to these big bloggers, maybe we're closing the door on big, important orders. Perhaps the entire company should pivot—maybe it's easier or more profitable to serve 100 large blogs than 1000 medium ones. But how do we know if we say "no?"

Then again, if we say "yes" we might really be screwed. If we can't provide them the human and technical service they expect, now we've hurt a blogger, we get bad press, and we've wasted a bunch of time. Or even worse, we hold on for dear life but it's extremely unprofitable, and now we have this expensive, time-consuming albatross around our necks.

So we've said "yes" by quoting high enough that we know for certain we will make good money on the deal, so much so that it will partially fund something else we want to do. Maybe that means a big new advertising campaign, or hiring another WordPress expert for our staff.

There was one especially large customer where we literally thought of it like this: This deal needs to be big enough to not only make a reasonable profit on the operating expenses, but pay for an entire developer's salary (assuming bootstrapped, put-in-elbow-grease-for-stock low salary), because we know this new customer will occupy a lot of that person's time, but all the remaining time we get "for free."

So we've given a lot of qualified yeses, and many were rejected. (P.S. Now we're able to say "yes" to those same bloggers, but that's because over time we've taken on bigger and bigger customers, and now a blogger with 30 million month page-views is something we know we can handle.)

At Smart Bear I used this principle yet again. Companies would fly me out to help them implement a peer code review process, which half the time actually meant that "management" wanted me to convince everyone else that code review was a good idea, and invent a process painless enough that they might actually do it.

From a business perspective, this was a poor use of my time. These folks had already bought our software, so it didn't sell more seats. When you counted a travel day on either side of the engagement, the time I lost could have been spent landing just one additional customer or make some important changes to the code, either of which almost certainly makes us more money.

Therefore, initially I just said "no." But again that's wrong. Eventually I said "yes," but the price was $2500/day *including travel days*, which for these sorts of engagements is unheard of. (Typically you get reimbursed for travel expenses but not paid for that time.)

This immediately cut out most trips, but some remained. On those trips I'd haul in $10,000 for a week of easy work, which I'd often combine with a long weekend with my wife. And anyway those people *really wanted me there*, which made the work that much more enjoyable.

So the principle is straightforward: Set the conditions of "yes" such that:

1. If they say “yes,” you’re happy because the terms or money are *so good*, it more than compensates for the distraction, perhaps funding the thing you really want to do.
2. If they say “no,” you’re happy because it wasn’t a great fit anyway; it’s not a worthwhile return on your time and effort.

So that’s the punch-line, but before you leave I’d like to over-emphasize the idea of “funding the thing you really want to do.”

This can take many forms, but it’s the single best way of figuring out *how* to qualify your “yes.” Examples:

- “Yes” if it pays for an entire additional person.
- “Yes” if this extends the runway of our startup by at least three months.
- “Yes” if it completely funds development we’d like to do anyway.
- “Yes” if it means one of the co-founders can quit her day-job.
- “Yes” if it will completely pay for three new marketing efforts.

Think of it like another form of funding. Funding is always a distraction from actually running your business, so the amount of money you get must be transformative to the business. Each of those bullet points *are* transformative, in that each has the potential to move your company from “hobby” to “real business.”

And if they say “no,” you’re fine with that, because it would have been a distraction which wouldn’t have moved the needle.