Harvard Business Review

THE BIG IDEA

Blitzscaling

**The chaotic, sometimes grueling path
to high-growth, high-impact entrepreneurship**
An Interview with Reid Hoffman by Tim Sullivan

ALING

THE CHAOTIC, SOMETIMES GRUELING PATH TO HIGH-GROWTH, HIGH-IMPACT ENTREPRENEURSHIP

INTERVIEW BY TIM SULLIVAN

Reid Hoffman is one of Silicon Valley's grown-ups. After helping to found PayPal, he moved on to found LinkedIn, in 2002, which has turned him into a billionaire. He was an early investor in Facebook and now serves as a partner at the venture capital firm Greylock. He's written two

books, *The Start-Up of You* (with Ben Casnocha) and *The Alliance: Managing Talent in the Networked Age* (with Casnocha and Chris Yeh).

In the fall of 2015, Hoffman began teaching a computer science class called Technology-Enabled Blitzscaling at Stanford University, his alma mater, with John Lilly (a partner at Greylock and formerly the CEO of Mozilla), Allen Blue (cofounder of LinkedIn), and Chris Yeh (cofounder of Allied Talent). In this edited interview with Tim Sullivan, the editorial director of HBR Press, Hoffman talks about the challenges, risks, and payoffs of blitzscaling.

HBR: Let's start with the basics. What is blitzscaling? **Hoffman:** Blitzscaling is what you do when you need to grow really, really quickly. It's the science and art of rapidly building out a company to serve a large and usually global market, with the goal of becoming the first mover at scale.

This is high-impact entrepreneurship. These kinds of companies always create a lot of the jobs and industries of the future. For example, Amazon essentially invented e-commerce. Today, it has over 150,000 employees and has created countless jobs at Amazon sellers and partners. Google revolutionized how we find information—it has over 60,000 employees and has created many more jobs at its AdWords and AdSense partners.

Why this focus on fast growth? We're in a networked age. And I don't mean only the internet. Globalization is a form of network. It adds networks of transport, commerce, payment, and information flows around the world. In such an environment, you have to move faster, because competition from anywhere on the globe may beat you to scale.

Software has a natural affinity with blitzscaling, because the marginal costs of serving any size market are virtually zero. The more that software becomes integral to all industries, the faster things will move. Throw in AI machine learning, and the loops get even faster. So we're going to see more blitzscaling. Not just a little more, but a lot more.

How did you settle on the term "blitzscaling"? It has some interesting associations. I have obvious hesitations about the World War II association with the term "blitzkrieg." However, the intellectual parallels are so close that it is very informative. Before blitzkrieg emerged as a military tactic, armies didn't advance beyond their supply lines, which limited their speed. The theory of the blitzkrieg was that if you carried only what you absolutely needed, you could move very, very fast, surprise your enemies, and win. Once you got halfway to your destination, you had to decide whether to turn back or to abandon the lines and go on. Once you made the decision to move forward, you were all in. You won big or lost big.

Blitzscaling adopts a similar perspective. If a start-up determines that it needs to move very fast, it will take on far more risk than a company going through the normal, rational process of scaling up.

This kind of speed is necessary for offensive and defensive reasons. Offensively, your business may require a certain scale to be valuable. LinkedIn wasn't valuable until millions of people joined our network. Marketplaces like eBay must have both buyers and sellers at scale. Payment businesses like PayPal and e-commerce businesses like Amazon have low margins, so they require very high volumes. Defensively, you want to scale faster than your competitors because the first to reach customers may own them, and the advantages of scale may lead

BLITZSCALING IS WHAT YOU DO WHEN YOU NEED TO GROW REALLY, REALLY QUICKLY.

you to a winner-takes-most position. And in a global environment, you may not necessarily be aware of who your competition really is.

Are there several dimensions to the idea of scale? There are three kinds of scale. People naturally focus on two of them: growing your revenues and growing your customer base. And of course, if you don't get those right, then nothing else matters. But very few businesses can succeed on those fronts without also scaling the organization. An organization's size and its ability to execute determine whether it can capture customers and revenue.

We see scale as a series of stages, based on orders of magnitude: A family-scale business can measure its employees in single digits; a tribe in tens; a village in hundreds; a city in thousands. A nation has more than 10,000 employees. These are estimates, not precise guides; a company often remains a family until around 15 employees, a tribe until around 150, and so on.

At each level, the way you run various functions—financing the company, hiring and onboarding employees, marketing the product, and so on—changes significantly. There aren't rules governing this when you're blitzscaling; you use heuristics instead—and by that I mean guidelines that help you make decisions and learn on the fly.

Organizational scale is more about the character of the company than it is an exact employee head count—things don't change drastically at exactly 150 employees. And you're not necessarily scaling each element of the firm at the same time or rate. You're more likely to focus first on customer service and sales than other functions. But even then, you'll have to blitzscale the other parts of the organization. So all along you really do need to be thinking about the company as a whole: How will you allocate your talent, and then how will you grow it? How will you hold on to your culture? How will you communicate? How will your competitive landscape shift?

When does a start-up begin to blitzscale? At the family scale, you're usually raising money and figuring out exactly what your product or service is. You most likely have not launched a product yet.

At the tribe scale, you're just starting to have a real company. It's fairly rare—not unheard of, but rare—for blitzscaling to start at this phase unless you have a runaway hit of a product: PayPal or Instagram,

for example. More typically, you've launched some version of the product or service, and you've homed in on your target market. But you're still not certain that the start-up can really scale massively. There's always some level of risk. You may decide not to scale at this stage, because you're not sure you have a product-market fit yet. Or you may decide to move ahead anyway, because you know you absolutely need to, for the offensive and defensive reasons we just talked about.

So the blitzscale process usually starts between the tribe and village scale. By then you've ironed out the product-market fit, you have some data, and you know what the competitive landscape looks like.

This is when the logic of blitzscaling becomes very clear. Once you begin to prove—to yourself and others—that there's an interesting category and a big market opportunity, you attract all kinds of competition. At the low end, other start-ups may be launching their own version of your product or service and trying to achieve scale in the market before you. At the high end, established brands are figuring out how to leverage their own assets to own part or all of your space.

A start-up has two advantages as a first mover going through blitzscale: focus and speed. Established brands tend not to be as fast or as focused. And competing start-ups probably don't have momentum yet (although they may be just as fast and focused).

The canonical example is Groupon, which made it to this middle stage and got hit by massive competition on both the high and the low ends. It wasn't able to both scale fast and build a durable product and thus failed to fully realize a potentially industry-transforming opportunity.

What organizational issues do you run into when blitzscaling? Blitzscaling is always managerially inefficient—and it burns through a lot of capital quickly. But you have to be willing to take on these inefficiencies in order to scale up. That's the opposite of what large organizations optimize for.

In hiring, for instance, you may need to get as many warm bodies through the door as possible, as quickly as you can—while hiring quality employees and maintaining the company culture. How do you do that? Different companies use different hacks. As part of blitzscaling at Uber, managers would ask a newly hired engineer, "Who are the three best engineers you've worked with in your previous job?"
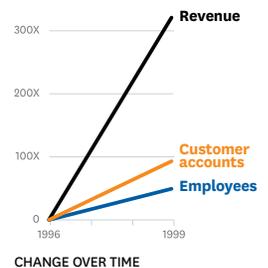
BY THE NUMBERS
AMAZON

EMPLOYEES
1996: **151**
1999: **7,600**

REVENUE
1996: **$5.1 MILLION**
1999: **$1.64 BILLION**

CUSTOMER ACCOUNTS
1996: **180,000**
1999: **16.9 MILLION**

CHANGE OVER TIME

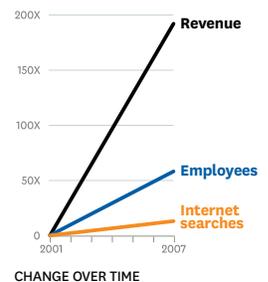BY THE NUMBERS
GOOGLE

EMPLOYEES
2001: **284**
2007: **16,805**

REVENUE
2001: **$86 MILLION**
2007: **$16.6 BILLION**

SEARCHES PER YEAR
2001: **27 BILLION**
2007: **372 BILLION**

CHANGE OVER TIME

ALL THE BANKING PEOPLE KNEW THE RULES. THAT PREVENTED THEM FROM TRYING ANYTHING THAT LOOKED REMOTELY LIKE PAYPAL.

And then they'd send those engineers offer letters. No interview. No reference checking. Just an offer letter. They've had to scale their engineering fast, and that's a key technique that they've deployed.

We faced this issue at PayPal. In early 2000, payment transaction volume was growing at a compounding rate of 2% to 5% per day. That kind of growth put PayPal in a deep hole as far as customer service was concerned. Even though the only place we listed our contact information was in the Palo Alto phone directory, angry customers were tracking down our main number and dialing extensions at random. Twenty-four hours a day, you could pick up literally any phone and talk to an angry customer.

So we turned off all our ringers and used our cell phones. But that wasn't a solution. We knew we needed to build a customer service capacity—fast.

But that's very difficult to do in Silicon Valley. So we decided to scale up in Omaha. This was during the first dot-com boom, so we convinced the governor of Nebraska that he wanted a piece of the internet revolution. He and the mayor held press conferences about how PayPal was going to open a customer service office, prompting a flood of job applicants. For four weekends straight, we flew out about 20% of the company to interview them. People showed up with their résumés, and we'd put them in a room and do group interviews. Within six weeks, we had 100 active customer-service people fielding e-mails.

It's now a classic technique for internet companies to offer e-mail and web-based customer service only. But we had to figure out how to hack our customer service challenge at a very fast pace. There was no playbook to tell us what to do. There still isn't.

If there are no rules, how do you come up with your approach? Sometimes freedom from normal rules is what gives you competitive advantage. For example, if we had understood how pernicious credit card fraud and chargebacks were in the early days at PayPal, I'm not sure we would have believed that such a service could be successful. We didn't realize how staggering the losses could be.

All the banking people knew the rules—you had to protect against fraud first. That prevented them from trying anything that looked remotely like PayPal. Our ignorance allowed us to build something fast, but then of course we had to fix it on the run, because we were already in the minefield.

Most critics thought we were losing so much money in 2000 because of our customer acquisition bonuses. But that wasn't the case. The industry's average customer-acquisition cost through advertising was around $40. So when we gave customers who recommended a friend 10 bucks and gave the new customer 10 bucks, we were cutting costs in half.

Why depend on heuristics rather than rules? Because you're looking for an edge that distinguishes you from other competitors, who are following conventional wisdom. That's not to say that there aren't rules. Don't allow anyone to embezzle your money. That's a rule. But it doesn't give anyone a competitive edge.

MARC OLIVIER LEBLANC

It sounds as if your choice of heuristics can lead to radically different organizational outcomes. Yes. One of the differentiators between Google and Microsoft, two blitzscaling companies, was that Google wanted to stay very flat, whereas Microsoft built up a lot of hierarchy.

You had to have eight direct reports at Google to be a manager, but there was no upper limit. People had 10, 15, 20, even 100 direct reports to minimize middle management. It would likely have been more managerially efficient to give someone no more than eight people. However, Google chose a flat organization that sacrificed that kind of efficiency to achieve an extreme focus on technology development. Microsoft, on the other hand, followed a more classical and hierarchical approach.

That reminds me of Google's decision to hire only people with very high GPAs from elite universities. As a heuristic, there's obviously collateral damage—there are many smart people you're not allowed to hire—but it makes sense if your goal is to hire a large number of smart generalists quickly. That created a lot of frustration. "I can't hire my friend who doesn't have that qualification, but I know that he's really good." And the company says, "Yeah, sorry. That's the way we execute as we blitzscale. We need a simple heuristic so that we can focus on what really matters." Another benefit of Google's decision to hire only from elite universities is that it helped create and maintain a coherent culture as the company scaled.

Why is culture so important to blitzscaling? Because you're growing an organization very fast, you have to make people accountable to each other on a horizontal or peer-to-peer basis, and not just vertically and top-down through the hierarchy.

What other heuristics are important as you go from, say, village to city? Specialization at all levels becomes more important. You need to understand how to run a large-scale engineering department, for example, and how to deploy a significant amount of capital in marketing. You need dashboards and analytics and metrics for those functions as much as you need them to help you understand customers and the marketplace.

You also need to have much higher reliability; sometimes the inefficiency that you accepted as you blitzscaled through the village stage is no longer tenable at a larger scale. You have to hire people who know how to make sure that your site is never down. And you have to be more careful in your release of engineering product. As a result, you have less adaptability. For example, Facebook famously shifted from a mantra of "Move fast and break things" to "Move fast with stable infrastructure."

You also move from a single-threaded organization to a multi-threaded one, allowing the company to focus on more than one thing at a time. When you're in a tribe, everybody is attuned to one priority. In a village, you're likely to start focusing on the thing that you're going to scale. You're also beginning to think about side experiments—for example, building developer tools, or experimenting with marketing or other paid acquisition. And you're likely adding new functions, like corporate development to consider acquisitions.

All of this rolls up to your macro goal of succeeding as a company, but as you move from village to city, functions are beginning to be differentiated; you're really multi-threading.

Companies at the city scale usually have more than one main product. They may have one central revenue stream, such as Google's AdWords or Microsoft Office, but several different products. They've built an architecture that determines how the products relate to each other. And each product can be multi-threaded as well.

Most Silicon Valley firms go global as they move from village to city, but some are global from Day One. At LinkedIn, we launched with 15 countries on our drop-down list. By the second day, we were getting e-mails from people whose countries were not on the list. It was an interesting geographic lesson for me, because I wasn't aware that the Faroe Islands was a country until we got a complaint. So I went and read a little history and said, OK, add it to the list. It's real.

Do different pockets of the company use different playbooks? Yes. For example, Google developed two device operating systems simultaneously: Android and Chrome. When Google acquired Andy Rubin and his start-up, Android Inc., Andy was set up as an entrepreneur within Google, focused on this experiment, and accountable to Larry Page. From Google's corporate resources perspective, it

---

BY THE NUMBERS

FACEBOOK

EMPLOYEES
2006: **150**
2011: **3,200**

REVENUE
2006: **$48 MILLION**
2011: **$3.7 BILLION**

ACTIVE USERS
(MONTHLY)
2006: **12 MILLION**
2011: **845 MILLION**

CHANGE OVER TIME
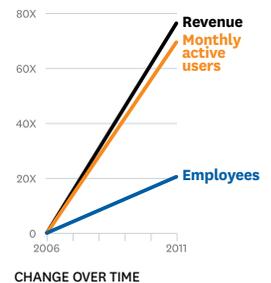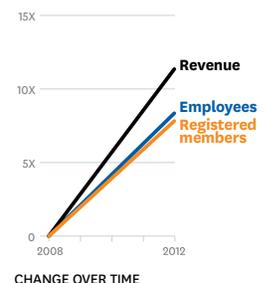

BY THE NUMBERS

LINKEDIN

EMPLOYEES
2008: **370**
2012: **3,458**

REVENUE
2008: **$79 MILLION**
2012: **$972 MILLION**

REGISTERED MEMBERS
2008: **17 MILLION**
2012: **150 MILLION**

CHANGE OVER TIME

April 2016 Harvard Business Review **7**

LEVELS OF ORGANIZATIONAL SCALE
A GLOSSARY

	Number of Employees	User Scale	Revenue
FAMILY	<10	<100K	<$10M
TRIBE	10–100	100K–1M	$10M+
VILLAGE	100–1,000	1M–10M	$100M+
CITY	1,000–10,000	10M–100M	$1B+
NATION	>10,000	>100M	$5B+

was a matter of asking Andy what he needed to make the project work.

Andy wanted Android to stay cohesive and focused. So for example, only Android employees' badges would grant access to the Android office; general Google employees couldn't get in. The Android team didn't run its software through Google's standard code review process. Andy also wanted to be able to cut different deals with mobile operators—whatever it took to get his project off the ground—without a cross-check.

In a completely different initiative, Chrome was developed in C++ (Android was developed in Java) and focused on laptops and browsers, rather than phones. Google could have handled that differently, by bundling Android and Chrome into one project, coherently attacking the device OS opportunity. But it chose instead to multi-thread, hiring the best person for the project, giving him the tools to get the job done, and letting him run a completely separate project and develop his own playbook.

One of the questions I've heard you ask is, What can you ignore? And maybe the flip side of that is, At each stage, what first-order problems are you solving? One of the metaphors that I use for start-ups is, you throw yourself off a cliff and assemble your airplane on the way down. If you don't solve the right problem at the right time, that's the end. Mortality puts priorities into sharp focus.

When you're blitzscaling, a whole bunch of things are inevitably broken, and you can't work on them all at once. You have to triage. You fix the things that will get investors to give you more cash.

The lift that capital provides means you have a longer time in the air to get things right. You're unlikely to get your plane to fly on your first capital lift or even your second.

A general principle of management is that if you have team dynamics problems, you fix them right away. But in blitzscaling, you're adding those challenges all the time. And you're moving so fast that today's problems aren't going to be the same as tomorrow's. The operation is always patched together and kind of ugly and held together with duct tape. So maybe you ignore the team's dysfunction for a while.

For example, your engineers might be unhappy. You think, Should we build development tools to help them be more productive? Should we allocate a bunch of our engineers to make that happen? But you know that the size of the team will continue to change radically; any tools you create today are going to be obsolete. So you don't try to solve that problem yet, even though you know that ignoring it will breed organizational unhappiness and that people will be frustrated. In nonblitzscaling circumstances those kinds of issues might be a top priority, but when you're blitzscaling, sometimes you have to just let them burn.

Remember, even if you do solve the problem, it will most likely stay solved only for a short time.

Can you alleviate unhappiness by telling people why you're making certain decisions? Yes, but only to a limited extent. What really keeps it all together is the perception that you're moving at high speed because you're growing something big, and that you're going to be part of something successful.

Almost every blitzscaling org that I have seen up close has a lot of internal unhappiness. Fuzziness about roles and responsibilities, unhappiness about the lack of a clearly defined sandbox to operate in. "Oh my God, it's chaos, this place is a mess." The thing that keeps these companies together—whether it's PayPal, Google, eBay, Facebook, LinkedIn, or Twitter—is the sense of excitement about what's happening and the vision of a great future. Because I'm part of a team that's doing something big, I'll work through my local unhappiness. Sure, I'd like a tidier sandbox, I'd like to be more efficient, I'd like the organization to be run more smoothly. But I'm willing to let it go because the pain will be worth it. ▽

HBR Reprint R1604B