

# ERIC SIMONS

LENNY'S PODCAST

BILINGUAL TRANSCRIPT

---

ORIGINAL BY

Lenny Rachitsky

@lennysan • x.com/lennysan

ANALYSIS BY

@Penny777 • x.com/penny777

# Eric Simons - 双语对照

This is the complete bilingual transcript for Lenny's Podcast featuring Eric Simons, CEO of StackBlitz (creators of Bolt).

---

## (00:00:00) Lenny Rachitsky

**English:**

The rate you're growing is absurd. You're in this cohort of companies that are just growing at rates that we've never seen in the history of startups.

**中文翻译:**

你们的增长速度简直荒谬。你们属于这样一类公司，其增长速度在初创企业历史上是前所未见的。

---

## (00:00:05) Eric Simons

**English:**

The company was on the verge of going under when we launched Bolt, and what ended up happening is, in the first two months it went from zero to 20 million of ARR. And we've already crossed 30 million of ARR, with the current rate we're on, our forecast for the year is we want to get to 100 million of ARR.

**中文翻译:**

在我们推出 Bolt 时，公司正处于破产边缘。结果发生的是，在头两个月里，我们的 ARR（年度经常性收入）从零增长到了 2000 万美元。现在我们已经突破了 3000 万美元的 ARR，按照目前的增速，我们今年的预测目标是达到 1 亿美元的 ARR。

---

## (00:00:22) Lenny Rachitsky

**English:**

This is just non-stop wild shit. How is this possible? What has allowed you to grow this much, this fast, with such a small team?

**中文翻译:**

这简直是接连不断的疯狂奇迹。这怎么可能呢？是什么让你们在团队规模如此之小的情况下，实现如此巨大且迅速的增长？

---

## (00:00:30) Eric Simons

**English:**

Most importantly, it's been the people. It's rare to find startups where you have the core group of five, six, seven people that have been there for five years plus.

**中文翻译:**

最重要的是人才。在初创公司中，很难见到五、六、七个人的核心团队能在一起共事五年以上，这非常罕见。

---

### (00:00:38) Lenny Rachitsky

**English:**

You basically were building a tech first, and then looking for a problem to solve later, which is often what people tell you not to do.

**中文翻译:**

你基本上是先开发技术，然后再寻找要解决的问题，而这通常是人们告诫你不要去做的事情。

---

### (00:00:44) Eric Simons

**English:**

I think that's the hard thing about being an entrepreneur. There are periods of time where you have to make judgment calls that are not going to be the consensus view. You got to have confidence in your convictions on how to best play the hand.

**中文翻译:**

我认为这就是做创业者最难的地方。在某些阶段，你必须做出一些非共识性的判断。你必须对自己的信念有信心，知道如何打好手里的这副牌。

---

### (00:00:54) Lenny Rachitsky

**English:**

A lot of people see these stats, and they sometimes don't see that there was also years and years of work before that.

**中文翻译:**

很多人只看到了这些数据，却往往忽略了在那之前长达数年的艰苦努力。

---

### (00:00:59) Eric Simons

**English:**

It was kind of like, Bolt's this overnight success, seven years in the making.

**中文翻译:**

这有点像，Bolt 是一个筹备了七年才实现的“一夜成名”。

---

### (00:01:05) Lenny Rachitsky

**English:**

Today my guest is Eric Simons. Eric is co-founder and CEO of StackBlitz, which makes a product called Bolt, which is currently neck and neck with Cursor for being the fastest growing product in history. They're currently the number one most popular web AI code app with over three million registered users. Two months after launching last October, they hit 20 million ARR. At the time of this recording, they're approaching 40 million ARR. The story of Bolt is wild. They actually started the company seven years ago, and were about to run out of money and shut down. But they realized the tech that they'd been building for the past seven years, called WebContainer, was perfectly suited for building AI products in a browser. So they launched the product with a tweet, and as Eric describes it, it was an overnight success seven years in the making. If you'd like to better understand the cutting edge of AI coding apps, and where things are going with AI and product building, this episode is a must listen.

#### 中文翻译:

今天的嘉宾是 Eric Simons。Eric 是 StackBlitz 的联合创始人兼 CEO，他们开发了一款名为 Bolt 的产品，目前正与 Cursor 旗鼓相当，争夺历史上增长最快产品的宝座。他们目前是全球最受欢迎的 Web AI 代码应用，拥有超过 300 万注册用户。去年 10 月发布两个月后，他们的 ARR 就达到了 2000 万美元。在录制本期节目时，他们的 ARR 已接近 4000 万美元。Bolt 的故事非常传奇。他们实际上在七年前就创立了公司，曾一度面临资金枯竭和倒闭。但他们意识到，过去七年一直在开发的名为 WebContainer 的技术，完美契合在浏览器中构建 AI 产品。于是，他们通过一条推文发布了产品，正如 Eric 所描述的，这是七年磨一剑的一夜成名。如果你想深入了解 AI 编程应用的尖端技术，以及 AI 与产品构建的未来走向，这一集绝对不容错过。

---

## (00:02:02) Lenny Rachitsky

#### English:

If you enjoy this podcast, don't forget to subscribe and follow it in your favorite podcasting app or YouTube. Also, if you become a yearly subscriber of my newsletter, you now get a year free of Perplexity Pro, Notion, Linear Granola, and Superhuman. Check it out at [lennysnewsletter.com](http://lennysnewsletter.com). With that, I bring you Eric Simons.

#### 中文翻译:

如果你喜欢这个播客，别忘了在常用的播客应用或 YouTube 上订阅和关注。此外，如果你成为我时事通讯（Newsletter）的年度订阅者，现在可以免费获得一年的 Perplexity Pro、Notion、Linear Granola 和 Superhuman。欢迎访问 [lennysnewsletter.com](http://lennysnewsletter.com) 查看。下面，让我们欢迎 Eric Simons。

---

## (00:02:22) Lenny Rachitsky (Ad Segment: Eppo)

#### English:

This episode is brought to you by Eppo. Eppo is a next generation A-B testing and feature management platform built by alums of Airbnb and Snowflake for modern growth teams. Companies like Twitch, Miro, ClickUp, and DraftKings rely on Eppo to power their experiments. Experimentation is increasingly essential for driving growth, and for understanding the performance of new features, and Eppo helps you increase experimentation velocity while unlocking rigorous deep analysis, in a way that no other commercial tool does. When I was at Airbnb, one of the things that I loved most was our experimentation platform where I could set up experiments, easily troubleshoot issues, and analyze performance all on my own. Eppo does all that and more, with advanced statistical methods that can help you shave weeks off experiment time, and accessible UI for diving deeper into performance, and out-of-the-box reporting that helps you avoid annoying, prolonged analytics cycles. Eppo also makes it easy for you to share experiment insights with your team, sparking new ideas for the A-B testing flywheel. Eppo powers experimentation across every use case, including product, growth, machine learning, monetization, and

email marketing. Check out Eppo at [geteppo.com/lenny](http://geteppo.com/lenny) and 10x your experiment velocity. That's get E-P-O .com slash lenny.

#### 中文翻译:

本集节目由 Eppo 赞助播出。Eppo 是由 Airbnb 和 Snowflake 的校友为现代增长团队打造的下一代 A/B 测试和功能管理平台。Twitch、Miro、ClickUp 和 DraftKings 等公司都依赖 Eppo 来驱动他们的实验。实验对于驱动增长和理解新功能表现变得越来越重要，Eppo 能帮助你提高实验速度，同时解锁严谨的深度分析，这是其他商业工具无法比拟的。我在 Airbnb 时，最喜欢的功能之一就是我们的实验平台，我可以独立设置实验、排查问题并分析表现。Eppo 不仅能做到这些，还拥有先进的统计方法，能帮你缩短数周的实验时间，并提供易用的 UI 进行深度性能分析，以及开箱即用的报告，帮你避免烦人且漫长的分析周期。Eppo 还能让你轻松与团队分享实验见解，激发 A/B 测试飞轮的新灵感。Eppo 支持各种场景的实验，包括产品、增长、机器学习、商业化和邮件营销。访问 [geteppo.com/lenny](http://geteppo.com/lenny) 了解 Eppo，让你的实验速度提升 10 倍。网址是 [get E-P-P-O .com/lenny](http://geteppo.com/lenny)。

---

### (00:03:41) Lenny Rachitsky (Ad Segment: Fundrise)

#### English:

This episode is brought to you by the Fundrise Flagship fund. Full disclosure, real estate investing is boring. Prediction markets are exciting, meme coins are a thrill ride, even the stock market can swing wildly on a headline. Hello, Deep Zeke. But with real estate investing, there's no drama, or adrenaline, or excuses to refresh your portfolio every few minutes. Just bland and boring stuff like diversification and dividends, so you won't be surprised to learn that the Fundrise Flagship Real Estate Fund is a complete snooze fest. The fund holds \$1.1 billion worth of institutional caliber real estate, managed by team of pros focused on steadily growing your net worth for decades to come. See? Boring. That's the point. You can start investing in minutes, and with as little as \$10 by visiting [fundrise.com/lenny](http://fundrise.com/lenny). Carefully consider the investment objectives, risks, charges, and expenses of the Fundrise Flagship Fund before investing. Find this information and more in the fund's prospectus at [fundrise.com/flagship](http://fundrise.com/flagship). This is a paid ad.

#### 中文翻译:

本集节目由 Fundrise Flagship 基金赞助。坦白说，房地产投资很无聊。预测市场很刺激，模因币（meme coins）像过山车，甚至股市也会因为一条新闻而剧烈波动。你好，Deep Zeke。但房地产投资没有戏剧性，没有肾上腺素，也没有理由让你每隔几分钟就刷新一次投资组合。只有平淡无奇的多样化和分红。所以你不会惊讶地发现，Fundrise Flagship 房地产基金简直让人昏昏欲睡。该基金持有价值 11 亿美元的机构级房地产，由专业团队管理，专注于在未来几十年稳步增长你的净资产。看吧？很无聊。但这正是重点。你可以在几分钟内开始投资，最低只需 10 美元，访问 [fundrise.com/lenny](http://fundrise.com/lenny) 即可。投资前请仔细考虑 Fundrise Flagship 基金的投资目标、风险、费用和支出。更多信息请参阅基金招股说明书：[fundrise.com/flagship](http://fundrise.com/flagship)。这是一则付费广告。

---

### (00:04:49) Lenny Rachitsky

#### English:

Eric, thank you so much for being here and welcome to the podcast.

#### 中文翻译:

Eric，非常感谢你能来，欢迎来到我们的播客。

---

### (00:04:52) Eric Simons

#### English:

Thank you for having me. Yeah, I'm stoked to be here.

**中文翻译:**

谢谢你的邀请。是的，能来这里我非常兴奋。

---

### (00:04:54) Lenny Rachitsky

**English:**

For folks that are not super familiar with Bolt, what is Bolt?

**中文翻译:**

对于那些不太熟悉 Bolt 的听众，你能介绍一下 Bolt 是什么吗？

---

### (00:04:58) Eric Simons

**English:**

It's really simple. You go there, there's a text box, and you tell it what you want to build. Whether it's a web or a mobile app, and so, it's kind of one of these text to app building tools that have become pretty popular over the past few months here. And it's not just building a static site, or something like that, but you can actually build full stack, real software with databases, and hosting and et cetera, just from prompting. And in a ridiculously short period of time, it's not like you're spending hours and hours or days, putting this together. You can get results in like, a minute.

**中文翻译:**

非常简单。你打开网页，看到一个文本框，然后告诉它你想构建什么。无论是 Web 应用还是移动应用，它就是过去几个月里变得非常流行的那种“文本生成应用”(text-to-app)的工具。而且它不仅仅是构建一个静态网站，你可以仅通过提示词 (prompting) 就构建出包含数据库、托管等功能的完整全栈真实软件。而且时间短得惊人，不需要花费数小时或数天，大约一分钟就能看到结果。

---

### (00:05:36) Lenny Rachitsky

**English:**

Let's just share some numbers about the scale of what you're building. The rate you're growing is absurd. You're kind in this cohort of companies that are just growing at rates that we've never seen in the history of startups, and you guys are at the edge of that. Share some numbers about how things went when you launched, and where they're at today.

**中文翻译:**

让我们分享一些关于你们规模的数据。你们的增长速度简直不可思议。你们属于那种在初创企业历史上从未见过的超高速增长公司之列，而且你们处于最前沿。分享一下发布时的情况以及现在的现状吧。

---

### (00:05:53) Eric Simons

**English:**

Yeah, when we launched, the company was on the verge of going under when we launched Bolt, our company StackBlitz. We'd been around for seven years building web-based development, environment

stuff. And so when we launched this we were like, "This would be amazing if this added a 100K of ARR over the next couple of months." And what ended up happening is, in the first two months, we went from zero to 20 million of ARR. And I think we're on month four, or four and a half or something like that at this point, and we've already crossed 30 million of ARR, and we're on the verge of crossing 40. By the time this comes out, it appears that we're going to be at 40 million ARR. So it's just, the scale of the growth of the revenue has been nuts. (00:06:41): And of course, that correlates with insane user growth, as well. We've added three million registered users just in the past few months here, and monthly active users is around a million, I think at this point, per month. So it's just been... I've never seen anything, I've been doing startups for 15 years, I've never seen anything like this. Everyone I've talked to, our investors, or et cetera, there's not a lot of corollaries to what's going on here. And it's kind of extraordinary, because our company wasn't doing AI stuff six months ago. We had no AI products, and just out of nowhere we, from almost death of the company to being the number one buy traffic revenue, et cetera, like AI Cogen app, that's totally web-based, in the world. I think the only other startup ahead of us is for Cogen, just in general, would be Cursor, and the option revenue at this point. (00:07:45): And so anyways, it's been a heck of a ride. And our team's like 15, 20 people, so it's just dealing with, we're going to be closing on 100,000 customers, and our support team's like three people. So we're trying to scale as fast as we can. So it's just kind of mind-boggling, just the scale of the demand, and how we've had to turn things around to match the demand as best as we can.

#### 中文翻译:

是的，当我们发布 Bolt 时，我们的公司 StackBlitz 正处于破产边缘。我们已经存在了七年，一直在做基于 Web 的开发环境。所以发布时我们想：“如果能在接下来的几个月里增加 10 万美元的 ARR，那就太棒了。”结果是，在头两个月里，我们的 ARR 从零飙升到了 2000 万美元。现在大概是第四个月或四个半月，我们已经突破了 3000 万美元的 ARR，正接近 4000 万美元。到这期节目播出时，我们可能已经达到 4000 万美元 ARR 了。所以，收入增长的规模简直疯了。当然，这与疯狂的用户增长也是相关的。过去几个月我们增加了 300 万注册用户，目前的月活跃用户（MAU）大约在 100 万左右。我做初创公司 15 年了，从未见过这样的景象。我交谈过的所有人，包括投资者，都觉得这事儿没多少先例可循。这非常不寻常，因为六个月前我们甚至还没开始做 AI 业务，没有任何 AI 产品。突然之间，我们从濒临倒闭变成了全球流量和收入排名第一的基于 Web 的 AI 代码生成应用。我认为在代码生成领域，唯一领先我们的初创公司可能就是 Cursor 了。总之，这是一段疯狂的旅程。我们的团队只有 15 到 20 人，却要应对接近 10 万名客户，而我们的支持团队只有 3 个人。所以我们正尽全力快速扩张。这种需求规模，以及我们为了匹配需求而进行的转型，简直令人难以置信。

---

## (00:08:14) Lenny Rachitsky

#### English:

Mind-boggling is an excellent way to describe what you just shared. A million monthly active users, you're at 40 million annual recurring revenue, five months into the business. Is that right?

#### 中文翻译:

“难以置信”确实是描述你刚才分享的内容的绝佳词汇。100 万月活用户，业务开展 5 个月 ARR 就达到 4000 万美元。我没听错吧？

---

## (00:08:26) Eric Simons

#### English:

Yeah, single digit. Yeah, single digit weeks. That's the current track rate that we're seeing for the thing. Yeah.

中文翻译:

是的，个位数。是的，仅仅几周时间。这是我们目前看到的运行速度。

---

### (00:08:33) Lenny Rachitsky

English:

I think, are you guys are the fastest growing startup in history?

中文翻译:

我想问，你们是历史上增长最快的初创公司吗？

---

### (00:08:37) Eric Simons

English:

I mean, I think it depends on probably where you peg the number. Because yeah, we're here to just build great products, and just push the limits of what's possible with the technology. And I think that we do our jobs well, kind of crazy things can happen, but I mean, the current track rate we have, we're going to be exceeding the forecast for Q1 with the current rate we're on, and our forecast for the year is we want to get to 100 million of ARR. And now, I think there's been a couple, that would either be on par with Cursor, or ahead of them, or something like that. (00:09:19): And I think there's going to be more things like this, too. I don't think that... It just, there's something really, I think a lot of people are in disbelief about it too, where they're like, "This is, okay." And this is from when we were at, got to four million ARR, five million ARR in the first month. I would talk with people and they're like, "Okay, yeah, but that could go to zero." And then it went to 20 the next, "Ah, it could go to zero," but now we're closing on 40. (00:09:40): So from my view, I was also very skeptical, as this. I've never seen anything grow like this, right? And so part of me, for like a month I was kind of waking up waiting for the day where it just was like, "Okay, it's over." You know what I mean? This crazy thing happened, and now it's not. But that data just hasn't come. And you see this happening with Cursor, you see this with a lot of these other AI startups. And the value proposition is real. The free market is filled with rational actors. People are coming to these tools because it is solving problems, they're able to do way more for way less cost, than it would otherwise. And that's why I said, I think we're going to see more of this, whether it's in coding or other verticals, or whatever. In a sense, it's almost like maybe the new normal, as AI just continues to get better. But, anyways.

中文翻译:

我想这取决于你如何界定这个数字。我们在这里只是为了打造伟大的产品，挑战技术的极限。如果我们把工作做好，疯狂的事情就会发生。按照目前的运行速度，我们将超过第一季度的预测，我们全年的目标是达到1亿美元的ARR。我想目前可能只有几家公司能达到这个水平，要么与Cursor持平，要么领先于他们。而且我认为未来还会出现更多这样的情况。很多人对此表示怀疑，比如当我们第一个月达到400万或500万ARR时，我跟人聊天，他们会说：“好吧，但这可能会归零。”然后下个月到了2000万，他们还是说：“啊，可能会归零。”但现在我们快到4000万了。从我的角度来看，我也曾非常怀疑。我从未见过增长如此之快的东西。有那么一个月，我每天醒来都在等那一天到来，就是那种“好吧，梦醒了，结束了”的感觉。你懂我的意思吗？这种疯狂的事情发生了，然后又消失了。但那种数据一直没出现。你在Cursor和很多其他AI初创公司身上也看到了这一点。价值主张是真实的。自由市场充满了理性人。人们使用这些工具是因为它们解决了问题，他们能以极低的成本完成更多工作。这就是为什么我说，我认为我们会看到更多这样的例子，无论是在编程还是其他垂直领域。在某种意义上，随着AI的不断进步，这几乎可能成为一种“新常态”。

---

## (00:10:41) Lenny Rachitsky

### English:

Let's get to a demo of Bolt, so people can actually see what this looks like in action. And as you go through it, if you can even point out stuff that is different from other products in the space, say Lovable, VZERO, Replit, that other folks have heard about, that'd be useful.

### 中文翻译:

让我们来看看 Bolt 的演示吧，这样大家就能亲眼看到它是如何运作的。在你演示的过程中，如果你能指出它与该领域其他产品（比如 Lovable、V0、Replit 等大家听过的工具）的不同之处，那就太有用了。

---

## (00:10:57) Eric Simons

### English:

Awesome. Cool. Yeah, so this is Bolt, you just go to bolt.new. Things that I think are really interesting about Bolt. One is, it's just dead simple. Whether you're logged in or logged out, it's the same UI, it's extremely simple, it's just a text box. And I think that the biggest difference between Bolt and the other stuff out there, it's actually subtle. It's not like something you'd necessarily see in the UI, but it's how fast it is, and how reliable it is. (00:11:23): And this is because of how we are actually doing the compute, because what's going on here is when you type into, whether it's Bolt or another product, it has to spin up a dev environment to actually make that application. So there needs to be some operating system somewhere that's running it. Everyone else runs those things on cloud servers, which those can take minutes to boot up, and they often will run into issues, and then you can end up literally stuck and have to contact support to get it done, and get it unstuck. (00:11:49): With Bolt, and for the past seven years, what our company's been doing has been building an operating system that runs inside of your browser locally, using your CPU. So we have a very permissive free tier, and it's insanely fast, and it's insanely reliable. (00:12:02): So if I want to, just as a quick example of this say, "Make a clone of Spotify," and just hit enter. This thing's already getting to work, and already, on the right here, this is a full dev environment. This is an actual operating system, running inside of my browser. And I can run commands on it, et cetera. And really, what you're seeing down here, this terminal and kind of what's backing it, this is what took us really five, six, seven years to build, and make so reliable. There would not be a Bolt without this technology called WebContainer, that allows us to run an operating system in the browser. (00:12:38): Because what's going on here is, our AI agent for Bolt has bidirectional communication with this operating system. It's writing code, it's running the dev server for this thing, it's going to go ahead and spin this up. You can see how fast this is, in a matter of 60 seconds I said, "Make me a Spotify clone," and now we have one. And it looks pretty darn good.

### 中文翻译:

太棒了。好的，这就是 Bolt，你只需要访问 bolt.new。我认为 Bolt 非常有趣的地方在于：第一，它极其简单。无论你是否登录，UI 都是一样的，非常简洁，就是一个文本框。我认为 Bolt 与其他产品最大的区别其实很微妙，不一定体现在 UI 上，而是它的速度和可靠性。这是因为我们处理计算的方式不同。当你输入指令时，无论是 Bolt 还是其他产品，都必须启动一个开发环境来构建应用。这意味着需要某个地方有一个操作系统在运行它。其他所有人都在云端服务器上运行这些环境，启动可能需要几分钟，而且经常会遇到问题，导致你卡住，不得不联系支持人员。而 Bolt 在过去的七年里，一直在构建一个直接在你浏览器本地运行、使用你本地 CPU 的操作系统。所以我们有一个非常宽松的免费层级，而且速度极快，极其可靠。举个简单的例子，我说“做一个 Spotify 的克隆版”，然后按回车。它已经开始工作了，右边就是一个完整的开发环境。这是一个真实的操作系统，运行在我的浏览器里。我可以在上面运行命令等等。你在这里看到的终端以及背后的支撑技术，就是我们花了五六七年时间才构建出来并使其如此可靠的东西。如果没有这项名为 WebContainer 的技术（允许我们在浏览器中运行操作系统），就不会有 Bolt。因为 Bolt 的 AI 智能体（agent）与这个操作系统有双向通信。它在

写代码，运行开发服务器，并启动它。你看这有多快，在 60 秒内，我说“给我做一个 Spotify 克隆版”，现在我们就得到了一个。而且看起来相当不错。

---

## (00:12:55) Lenny Rachitsky

**English:**

That looks really good.

**中文翻译:**

看起来真的很棒。

---

## (00:12:58) Eric Simons

**English:**

And that's one of the other aspects around Bolt is, this technology we made for the operating system side, the guys that have been working with us for the past five-plus years on it, before this they were actually doing machine learning AI stuff. And so when it came time to write the agent for Bolt, we had just an incredible amount of in-house expertise on how to actually merge these two different technology sets, to have this really reliable experience that produces really beautiful, really functional stuff. So that's based on what's really cool about the Bolt experience. (00:13:32): The other thing is, a lot of these products it's like, you can make something, but often you want to actually have a URL where you can share this. Having, maybe even attach a domain to it, or whatever have you. So with Bolt, we actually have built-in integrations with production grade hosting providers like Netlify, and for databases with Supabase. (00:13:53): So if I go and just click the deploy button here, this is actually going to run a production build of this project we made here. And again, this is doing this entirely inside of my browser, so it doesn't cost us anything to do this. So again, you can do this for free, and it has gone ahead and deployed this on a real URL, on Netlify. This is live, I can share this with anyone, and if I want to buy the domain spotifyclone.com, and point it at this, I can click this link here. That will kick me into Netlify, I can attach this to my account, buy a domain, point at that thing. And then from there on out, whenever I'm prompting Bolt to make changes to this application and hit deploy, that goes live on my public website. (00:14:36): So this is the simplest way to build a web app that's ever existed. That was one of the key realizations I had, a couple of weeks into the thing, I was seeing people use this for personal use cases. Like medical donation sites, or weddings, or whatever. And I was like, "Don't these people know that Wix or Squarespace exists? Should I tell them?" And then it hit me. Those things are so complicated to use. I don't know if you've ever seen just the UI of these things, but they're crazy complicated, and that's just for building a static website. There's no way you could actually build a functional app. And that's like, with Bolt, if we were to sit here for another 30 minutes, we would have streaming. You'd be able to make playlists of different MP3 files, or whatever. You can just keep prompting this thing to keep adding functionality. (00:15:25): So that's, I think, some of the cool core experience of both here. I can show you something cool that we just launched, if that would be of interest.

**中文翻译:**

这是 Bolt 的另一个方面：我们为操作系统端开发的技术，以及过去五年多一直和我们一起工作的团队，他们之前其实是做机器学习 AI 的。所以当需要为 Bolt 编写智能体时，我们在如何融合这两个不同的技术栈方面拥有惊人的内部专业知识，从而实现了这种非常可靠、能生成精美且功能齐全产品的体验。这就是 Bolt 体验中非常酷的一点。另一件事是，很多同类产品只能让你做点东西，但你通常想要一个可以分享的 URL，甚至绑定一个域名。在 Bolt 中，我们内置了与生产级托管商（如 Netlify）和数据库（如 Supabase）的集成。如果我点击这里的“部署”按钮，它实际上会为我们刚才做的项目运行一个生产环境构建。再次强调，这完全是在我的浏览器

内部完成的，所以对我们来说没有任何成本。你可以免费完成这一切，它已经在 Netlify 上部署了一个真实的 URL。这是实时的，我可以分享给任何人。如果我想买个 spotifyclone.com 的域名指向它，点击这个链接就能进入 Netlify 绑定账号。从那以后，只要我通过提示词让 Bolt 修改应用并点击部署，修改就会在我的公开网站上生效。这是有史以来构建 Web 应用最简单的方式。在项目开始几周后，我意识到一个关键点：我看到人们用它做个人用途，比如医疗捐赠网站、婚礼网站等。我当时想：“这些人不知道 Wix 或 Squarespace 吗？我要告诉他们吗？”然后我突然意识到，那些东西太难用了。如果你看过那些工具的 UI，你会发现它们复杂得离谱，而且那还只是为了建一个静态网站。你根本无法用它们构建一个功能性的应用。而用 Bolt，如果我们再坐在这里聊 30 分钟，我们就能做出流媒体功能，你能创建不同 MP3 文件的播放列表。你可以不断通过提示词来增加功能。这就是 Bolt 的核心体验。如果你感兴趣，我可以展示一下我们刚刚推出的酷功能。

---

## (00:15:32) Lenny Rachitsky

### English:

Let's do it.

### 中文翻译:

来吧。

---

## (00:15:32) Eric Simons

### English:

So this is like web apps, right? Web apps are amazing, but often you want to have a native app. And it's hard to build web apps, it's even harder to build native apps, that can actually, that you can then go put in the app store. And so we partnered up with a company called Expo, and their entire business is making, basically, React Native tooling and this ecosystem that makes it super easy to build beautiful apps, and actually get them in the app store. (00:15:57): And so right here, I'll zoom in a little bit, we have this little, "Build a mobile app with Expo." So if you click that, we kind of instruct you on how to just prompt mobile apps into existence. So yeah, let's make another Spotify clone that's an actual native mobile app. Let's say, "Make a Spotify clone," go ahead and hit enter. And what this thing's going to do is actually, again, spin up a operating system here, where it's going to boot up the Expo tool chain and actually go and make a mobile app for us. (00:16:28): And what's cool about this is, we could actually preview it just in the browser here, but once this thing's done and it boots up, it's going to show a QR code, we're going to be able to scan it, and in real time actually basically have a test flight of this native application that we can try it on our phones, and as we keep prompting you'll see it making changes and stuff. This is kind of the first time that, you don't have to be technical to make production grade web, full-stack web and mobile apps. At this point I've done nothing that requires developer knowledge to do any of this stuff. (00:17:10): And I think that's what a lot of people are really excited about with this, and you know, the majority of our audience are people that are not developers, that are using this. They're PMs, they're designers, they're entrepreneurs. Because these are people that have always been great at building products, but previously, the only way that they could get their ideas into coded software was through a developer's fingertips. And now, they can deal with their own, through prompting. (00:17:37): So you can see here, we've got this little QR code. I'm going to go ahead and scan the thing.

### 中文翻译:

刚才那是 Web 应用。Web 应用很棒，但你通常想要一个原生应用（native app）。构建 Web 应用很难，构建能上架应用商店的原生应用更难。于是我们与一家名为 Expo 的公司合作，他们的核心业务就是做 React Native 工具链和生态系统，让构建精美应用并上架商店变得超级简单。就在这里，我放大一点，有一个“用 Expo 构建移动应用”的小按钮。点击它，我们就会指导你如何通过提示词“变”出一个移动应用。好，让我们再做一个

Spotify 克隆版，但这次是真正的原生移动应用。输入“做一个 Spotify 克隆版”，按回车。它会再次在这里启动一个操作系统，加载 Expo 工具链，为我们构建移动应用。酷的地方在于，我们可以在浏览器里预览，但一旦它启动完成，会显示一个二维码。我们可以扫描它，实时在手机上进行“TestFlight”式的体验。随着我们不断输入提示词，你会看到它在实时修改。这是第一次，你不需要技术背景就能制作生产级的全栈 Web 和移动应用。到目前为止，我做的所有操作都不需要任何开发者知识。我认为这就是让很多人兴奋的地方。我们的大多数用户其实不是开发者，而是产品经理（PM）、设计师和创业者。因为这些人一直擅长打造产品，但以前他们只能通过开发者的手指将想法转化为代码软件。现在，他们可以通过提示词自己动手了。你看，这里有个二维码，我来扫一下。

---

### (00:17:43) Lenny Rachitsky

**English:**

I'm going to do it, too.

**中文翻译:**

我也来扫一下。

---

### (00:17:43) Eric Simons

**English:**

Cool.

**中文翻译:**

酷。

---

### (00:17:44) Lenny Rachitsky

**English:**

By the way, I love that you had just enough things to say until it finished. That was pro.

**中文翻译:**

顺便说一下，我太喜欢你刚好在它完成前把话说完的节奏了，太专业了。

---

### (00:17:50) Eric Simons

**English:**

Just as I planned, you know? So on my screen it's booting up, it's bundling the JavaScript of this thing, and it's beta. We just launched this last week, by the way. So if you can kind of see on my screen here, I actually have this Spotify looking app, right?

**中文翻译:**

正如我计划的那样。现在我的屏幕上正在启动，正在打包 JavaScript。顺便说一下，这是测试版，我们上周才发布的。如果你能看到我的屏幕，我这里已经有一个看起来像 Spotify 的应用了，对吧？

---

### (00:18:07) Lenny Rachitsky

**English:**

Wow.

**中文翻译:**

哇。

---

**(00:18:07) Eric Simons**

**English:**

That, you know.

**中文翻译:**

就是那个。

---

**(00:18:09) Lenny Rachitsky**

**English:**

That looks like, exactly like Spotify.

**中文翻译:**

看起来简直和 Spotify 一模一样。

---

**(00:18:11) Eric Simons**

**English:**

It looks exactly like Spotify, right?

**中文翻译:**

一模一样，对吧？

---

**(00:18:11) Lenny Rachitsky**

**English:**

It's good.

**中文翻译:**

太棒了。

---

**(00:18:11) Eric Simons**

**English:**

Yeah.

**中文翻译:**

是的。

## (00:18:14) Lenny Rachitsky

### English:

We're going to be sued right now, so let's be... You're doing too good a job with this. No, that's amazing.

### 中文翻译:

我们要被起诉了，所以我们还是……你做得太逼真了。不，这真的很神奇。

---

## (00:18:24) Eric Simons

### English:

Yeah. So it's pretty cool, right? And so what's cool is that, and as you keep prompting on your device, it'll just keep reloading. Without you having to kill the app, you can actually see the functionality getting added. And so, in this use case that you and I have right now, it's like if you and I were building an app together, we could be on other sides of the planet and you could actually be not just seeing a screenshot of the thing, but actually touching it and feeling it, and putting it through its paces. (00:18:47): And so a lot of product teams, I mean, this is just changing how people do product development. It's faster to do this than design a whole bunch of Figma frames, necessarily. Right? So.

### 中文翻译:

是的，很酷吧？更酷的是，当你继续在设备上输入提示词时，它会自动重新加载。你不需要关闭应用，就能看到新功能的加入。就像现在我们这个场景，如果我们一起开发应用，即使我们在地球的两端，你也不仅仅是看一张截图，而是能真实地触摸、感受并测试它。所以对于很多产品团队来说，这正在改变产品开发的方式。这样做有时比设计一堆 Figma 原型图还要快。对吧？

---

## (00:18:59) Lenny Rachitsky

### English:

We're going to spend a lot of time on that. Okay, this is incredible, this whole episode so far is you just blowing my mind and I imagine listeners' minds, just over and over and over. I don't even know where to go with all this, sometimes. (00:19:09): You made a really important point, that you worked on this for seven years before you launched Bolt. A lot of people see these stats, zero to 40 million ARR in five-ish months, and they sometimes don't see that there was also years and years of work before that. And the reason that you guys have been so successful is all the work you did that allowed, that built this WebContainer technology, it sounds like. Is there anything there that's worth sharing, you think, of just that part of the journey? I know we'll go through the origin that all, where Bolt came from, but I guess just that WebContainer component specifically. That feels like a huge deal.

### 中文翻译:

我们会花很多时间讨论这个。好的，这太不可思议了，到目前为止，你一直在不断地刷新我的认知，我想听众们也是一样。有时候我甚至不知道该从哪里开始聊了。你提到了一个非常重要的点：在发布 Bolt 之前，你们已经为此努力了七年。很多人看到这些数据——五个月内 ARR 从零增长到 4000 万美元——却往往忽略了背后的多年积累。听起来，你们之所以如此成功，是因为你们构建了 WebContainer 技术。关于那段旅程，你觉得有什么值得分享的吗？我知道我们会聊到 Bolt 的起源，但我想先聊聊 WebContainer 这个组件，这感觉是个大利器。

---

**English:**

A hundred percent it is, yeah. And I would say this is, surprisingly to me, it's still one of the contrarian viewpoints of our company. Because over the years it was like, when we first... And that, the WebContainer was the bet, that we made the company on. Just to be clear. StackBlitz was a browser-based, deep technology play on, "Can we make a web assembly based operating system that can boot in a browser, in like a hundred milliseconds, and run full on development tool chains?" That was really it. (00:20:21): And we'd gotten the idea for this, and the insight that this might be possible, because back when my co-founder and I came out to the Valley, he and I grew up down the street from each other in Chicago, we wrote code together at 13, we came out to the Valley in 2012, and we just had the good fortune of bumping into Dylan Field and Evan Wallace when they were building Figma, in the early days. And that was, I don't think a lot of people know that Figma was also a browser-based deep technology play. Their first pitch for Figma, they didn't have a design tool. Their first pitch was this 3D ball dropping into water, inside of a browser town. (00:21:00): And the pitch basically was, "Browsers have this new capability called WebGL," the predecessor to WebAssembly, "and with these things, for the first time, you could actually create a graphics rendering engine, that you could then build a design tool on top of. But you're going to have to write that rendering engine from scratch, because nothing exists that can just compile into WebGL, or whatever. And if you want the performance you need, et cetera, it's going to take us years to do, but if we do it, we think this will change everything for design." (00:21:35): And obviously, we know how that story has panned out now. And back in 2017, 2016, 2017, Albert, my co-founder and I, saw the same sort of story begin to play out, but for web development and development environments. And specifically there was some stuff that landed in browsers like WebAssembly, shared memory, service workers, these different APIs. And we were like, "Oh, wow. It should be possible, theoretically, to write an operating system in WebAssembly that could run Node.js, and NPM and all the tool chains on top of it, that you need to do web development." (00:22:12): And that would be huge, because setting up developer environments, it's a pain for beginners. A lot of people churn out. The first thing you do when you learn how to code is not even learning how to code, it's how to set up your computer to even start writing the code. If you go join Netflix, or any of these other fan companies, the first month or two is you being onboarded, to run that stuff on your computer and set up your environment. And we're like, "If we could just have that be something, you click a link and it just boots in your browser, that'd be huge." (00:22:42): It's also, if you look at the other productivity apps that have really worked on the web, they've all had this compute model, right? Figma, when you open a Figma document, there's not like some cloud VM that gets spun up for you to render the documents. You're dragging things around. It's using your CPU and your memory to do the work. Same thing with Google Docs. That's the only model that's ever scaled to a billion users. And so, when you look at Cloud IDEs, like Cloud 9 was the first one, back in 2009 or so. The way these have always worked is that your browser's basically doing nothing, when you go to that. Every user that gets connected, there has to be a cloud VM that gets spun up for them, and then your browser's just taking your keystrokes, sending it to the server, and then sending back the results of it. And that's how all these other AI code, text to app sort of tools work. They're all using cloud VMs. (00:23:34): And the problem is, on a small scale it can work, but as you scale it up, I mean there's not even a 100 million VMs to rent, on the planet. But there are a billion devices that you can run this stuff on. Because that's kind of what we've seen with Bolt where, if you want to build a product that's going to be able to scale to that size, you have to look at all factors and go, "We have to build, make sure the technology provides the best experience, zero latency, transient cost." There's a permissive free tier, because the other problem with the server is, you end up, if you have a free tier, people are mining Bitcoin on it, they're DDoSing people using your servers. So inevitably, you have to nerf these things and roll them back. But if it's all done on the end device, it doesn't matter. (00:24:18): So anyways, WebContainer was the key piece, and what we struggled with, it took us four or five years or something, to build

WebContainer. What we struggled with for the years after that was just how to build a product around it, because developers loved it, but they weren't using it in ways that they would pay money for. And as much as the nerd side of me wished that that would be enough, that it was like, building cool technology was enough. It's like, "It's not. We're here to build a venture scale company." And so that was kind of why we were high at the end of the journey, where it was like, we're taking shots on goal. And at some point, this got a connected bat, right?

#### 中文翻译:

百分之百是。而且我想说，令我惊讶的是，这仍然是我们公司最受争议的观点之一。因为多年来，WebContainer 一直是我们押注整个公司的赌注。StackBlitz 是一个基于浏览器的深度技术博弈，核心问题是：“我们能否制作一个基于 WebAssembly 的操作系统，让它在浏览器中 100 毫秒内启动，并运行完整的开发工具链？”这就是我们的初衷。我们之所以产生这个想法并洞察到其可能性，是因为我和我的联合创始人来到硅谷时（我们从小在芝加哥同一条街长大，13 岁一起写代码），我们在 2012 年来到硅谷，有幸在 Figma 早期遇到了 Dylan Field 和 Evan Wallace。我想很多人不知道 Figma 最初也是一个基于浏览器的深度技术博弈。他们第一次推销 Figma 时，甚至还没有设计工具，而是一个 3D 球掉进水里的浏览器演示。他们的逻辑是：“浏览器现在有了 WebGL（WebAssembly 的前身），利用它，你第一次可以创建图形渲染引擎，然后在其上构建设计工具。但你必须从头编写渲染引擎，因为当时没有任何东西能直接编译成 WebGL。虽然这需要数年时间，但如果成功，它将改变设计界的一切。”显然，我们现在知道那个故事的结果了。在 2016、2017 年左右，我和联合创始人 Albert 看到同样的故事在 Web 开发和开发环境领域上演。当时浏览器出现了一些新 API，如 WebAssembly、共享内存、Service Workers。我们觉得，理论上应该可以在 WebAssembly 中编写一个操作系统，运行 Node.js、NPM 以及所有 Web 开发所需的工具链。这将是巨大的突破，因为配置开发环境对初学者来说非常痛苦，很多人因此放弃。学习编程的第一件事甚至不是编程，而是如何配置电脑。如果你加入 Netflix 这样的公司，头一两个月都在熟悉环境。我们想，如果只需点击一个链接就能在浏览器中启动环境，那该多棒。此外，如果你看那些在 Web 上真正成功的生产力应用，它们都有这种计算模型。比如 Figma，当你打开文档时，并没有云端虚拟机（VM）为你渲染，而是利用你本地的 CPU 和内存。Google Docs 也是如此。这是唯一能扩展到 100 亿用户的模型。而早期的云端 IDE（如 2009 年的 Cloud 9），浏览器基本不干活，每个用户连接都要启动一个云端虚拟机，浏览器只是把按键传给服务器再传回结果。现在的其他 AI 代码工具也是这么干的。问题是，小规模可行，但大规模扩展时，地球上甚至没有 1 亿个虚拟机可供租用，但却有 10 亿台设备可以运行这些东西。这就是我们在 Bolt 身上看到的：如果你想构建一个能扩展到那种规模的产品，你必须确保技术提供最佳体验、零延迟和极低成本。我们可以提供宽松的免费层级，因为如果是服务器端，免费用户可能会用来挖矿或发起 DDoS 攻击，你不得不限制他们。但如果在终端设备上运行，就没关系了。总之，WebContainer 是关键。我们花了四五年时间才把它做出来。之后几年的挑战是如何围绕它打造产品，因为开发者虽然喜欢它，但还没到愿意为此付钱的地步。我内心的极客部分希望“酷技术”就足够了，但事实并非如此，我们要建立的是一家风险投资规模的公司。所以我们不断尝试，直到最后终于“击中了球”。

---

## (00:25:04) Lenny Rachitsky

#### English:

There's a lot of really interesting lessons from this journey, that I think are counterintuitive. One is, you basically were building a tech first, and then looking for a problem to solve later. Which is often what people tell you not to do. And it worked out, in this case. (00:25:19): The other interesting takeaway here is, it feels like it's a similar moment to when AJAX came out and then everyone's just like, "Wow, you can build new things here." So it feels like there's a lesson here of just, "If there's a new technology that has enabled, something big that we think may, let's just work there for a while, and see if something comes up." (00:25:35): And then I think the other lesson here is just, as a founder, just survive as long as you can. Because you may find something that works.

#### 中文翻译:

这段旅程中有很多非常有趣的教训，而且我认为很多是反直觉的。第一，你基本上是先开发技术，然后再寻找要解决的问题。这通常是人们告诫你不要去做的事，但在这种情况下它成功了。另一个有趣的感悟是，这感觉就像当年 AJAX 出现时的那一刻，每个人都惊叹：“哇，原来可以在这里构建新东西。”所以这里的教训是：“如果出现了一项可能开启大机会的新技术，那就先在那里深耕一段时间，看看会发生什么。”第三个教训是，作为创始人，只要尽可能久地生存下去。因为你最终可能会发现行之有效的东西。

---

## (00:25:43) Eric Simons

### English:

All great points, all great points. Because you're dead right. And fortunately, my co-founder and I had, we had built a lot of unsuccessful stars before this. We spend most of the, or 20 times, churning through ideas on things. So when we had conviction, I was like, "This seems like a technology that will be important." It seems like, the web is the most ubiquitous... The pitch or the theory in our head was like, "The web is the most ubiquitous platform in the world, but yet it has no, you can't use the web to build the web." (00:26:08): Every other platform, Mac has Xcode. Windows has Visual Studio. The web had nothing. And we were like, "At a minimum, Google should probably buy this thing from us. It seems like it should probably be part of Chrome," at a minimum. And we thought, "Hey, this could be a huge enabler." The vision of just making it as easy to build full stack applications as using Canva, it just seemed really compelling. (00:26:43): But when you do that sort of risky deep technology play, you need to... And we were very good about this, like the previous company Albert and I did, we bootstrapped it all the way through to acquisition, so we understood and we were living hand-to-mouth, to bootstrap that thing. So we understood out of it how to have a low burn rate, and take a lot of shots on goal, and make every dollar stretch beyond what anyone would think is reasonable or possible. And that's how we played our hands with StackBlitz. We didn't raise money for the first two or three years of the company's life. We were bootstrapping it. When we did raise money, we barely spent it. Largely because it was like, "We need to just take a lot of smart bets, and it doesn't make sense." (00:27:32): And I would just say generally, until you see pull, just people pulling the product out of your hands, you don't want to be spending money. You should be like, default, no. And when you go and buy software, you should be going, "We're a tiny startup. Can you sell it for half?" Everything you buy, just keep the burn rate as low as possible, because you need as many shots on goal as you can possibly get. Because you have no idea. I think just generally, for startups, that's the right way in my view, to approach it. Unless you're seeing, again, immediate demand and pull, or whatever. (00:28:02): But yeah, I think that'd be, maybe the extra context I'd add on top is, I think that we ended up doing a good job of being extremely conservative. During a time in which, during 2020, through 2020 and 2021, which were times where exuberance and growing headcount was like, KPIs of companies. And were things that were being... With lot of emotional force of like, "Hey, you guys ought to be doing this." And I'm glad that we didn't heed the advice, because if we had tripled the company and kicked up the burn rate, there would be no Bolt. We would've gone out of business a lot of time ago. (00:28:48): So I think that's the hard thing about being an entrepreneur, I think is you kind of have to... There are periods of time where you have to make judgment calls that are not going to be the consensus view. Maybe years later, it'll become the consensus view, but you got to have confidence in your convictions on how to best play the hand.

### 中文翻译:

说得太对了。因为你完全正确。幸运的是，我和联合创始人在此之前已经尝试过很多不成功的项目。我们花了大量时间在各种想法中摸爬滚打。当我们产生信念时，我想：“这看起来是一项非常重要的技术。” Web 是世界上最普及的平台，但你却不能用 Web 来构建 Web。其他平台都有自己的工具，Mac 有 Xcode，Windows 有 Visual Studio，Web 却一无所有。我们当时想：“最起码，Google 应该会买下这东西，它应该是 Chrome 的一部分。” 我们认为这会是一个巨大的赋能工具。让构建全栈应用变得像使用 Canva 一样简单，这个愿景非常

吸引人。但当你进行这种高风险的深度技术博弈时，你必须……我们在这方面做得很好，因为我和 Albert 之前的公司是完全靠自筹资金 (bootstrap) 直到被收购的，所以我们习惯了精打细算。我们知道如何保持低烧钱率 (burn rate)，多尝试，让每一块钱都发挥出超乎想象的作用。我们在 StackBlitz 也是这么做的。公司成立的前两三年我们没拿融资，全是自筹。即使后来拿了钱，我们也几乎没花。主要是因为我们觉得：“我们需要进行很多聪明的尝试，大肆挥霍没有意义。”我想说，在看到真正的“拉动力” (pull)，即人们抢着要你的产品之前，你不应该乱花钱。你的默认态度应该是“不”。当你买软件时，你应该说：“我们是家小创业公司，能半价卖给我们吗？”尽可能降低烧钱率，因为你需要尽可能多的尝试机会。你永远不知道哪次会成功。我认为对于初创公司来说，这是正确的方法，除非你看到了即时的爆发性需求。我想补充的背景是，我们在保持极端保守方面做得很好。在 2020 年和 2021 年那个狂热的时期，增加员工人数被视为公司的 KPI，很多人都在情感上施压说：“嘿，你们应该扩张。”我很高兴我们没听那些建议，因为如果我们当时把公司规模扩大三倍并提高烧钱率，就不会有今天的 Bolt 了，我们早就倒闭了。这就是做创业者最难的地方：你必须在某些时刻做出非共识性的判断。也许几年后它会成为共识，但你必须对自己的信念有信心。

---

### (00:29:15) Lenny Rachitsky

#### English:

There's so many great lessons here, I think just this idea of just staying alive. Dalton came on the podcast, he's a partner at YC Ones, and he just had this phrase, "Just don't die." And that's exactly what you guys did, seven years of just trying it until something worked, and I love that you actually were planning to shut down the company right before you launched Bolt. And I know you launched it with just a tweet, right? That was the launch moment?

#### 中文翻译:

这里有很多伟大的教训，尤其是“活下去”这个理念。YC 的合伙人 Dalton 曾来过这个播客，他有一句名言：“只要不死就行。”你们正是这么做的，七年间不断尝试直到成功。我特别喜欢的一点是，在发布 Bolt 之前，你们其实已经打算关闭公司了。而且我知道你们发布它只用了一条推文，对吧？那就是发布的时刻？

---

### (00:29:35) Eric Simons

#### English:

Yep. Yeah.

#### 中文翻译:

是的。

---

### (00:29:37) Lenny Rachitsky

#### English:

Maybe talk about that moment of just, after launch, signs that, "Okay, this is working. Something's different."

#### 中文翻译:

聊聊那个时刻吧，发布之后，有哪些迹象让你觉得：“好吧，这行得通，情况不一样了。”

---

### (00:29:43) Eric Simons

#### English:

Yeah, yeah. So day one it was like, there's great reception to the tweet. We were like, "Wow, this is one of the biggest things, launch day reception we've ever seen." And I think on the first day, I think we added 60K of ARR, or something. Which was like, I mean, crazy. Again, we were at 600, so we added 10% in a day. And I remember our dev ops engineer, he was the one who would flag me. He was like, "Guys, we got 60K today. This is crazy." And I was like, "Yeah, yeah. But this is launch day." (00:30:12): There's the tech crunch, peak of initiation, in the classic startup- (00:30:17): [Inaudible 00:30:17]- (00:30:17): ... star, yeah. I was like, "Listen, guys." I'm trying to temper enthusiasm for the team. I'm like, "This is great. Got a lot of work to do." And then the next day we added 80K, or whatever it was, and it just kind of kept going. And all the while, the product we put out, we built a thing in 90 days. We built Bolt in 90. So there's a lot of things that were missing in the product. Like, basic stuff, basic stuff. And which, again, we cut the right corners on the thing to get it online, but we had this just growing influx of people using it, going, "How is there not a mobile responsive view? How are chat messages not," we got to 20 million of ARR without a mobile responsive view, by the way. Just throwing that out there. It was like the iPhone not having copy and paste until iPhone 5, or whatever. That was that, this was that for us, it was like, no mobile. You looked at it on mobile, it was terrible. (00:31:13): But there was stuff like that, so we had to just... And then, we're a small team and so, we were completely unprepared for just the growing traffic. And there was a whole bunch, I mean, the list of problems that were happening every single day was nuts. I mean, to start, we had never had a plan on stackblitz.com for more than \$9.00. We had one price, nine bucks. And so when we launched Bolt we were like, "Again, we don't think, hopefully people like this, but nine bucks doesn't get you a lot of inference." And so people burn through nine bucks in 48 hours. And they're like, "I want to buy more. How do I buy more? Why won't you take my money?" (00:31:56): So it was like, within the week we rolled out just completely new pricing plans, where you could upgrade, which ended up, has kind of now become the standard. All the other guys in the space have copied this. Where prior to Bolt going online, Copilot, all these previous AI things, everyone wanted this Netflix model where there's one price, it's like all you can eat, or whatever. And the problem is, if you do that, you want the inference cost to be kind of low, because you're expecting people to use it a lot. And so you can't do these agentic experience things, it would be too expensive. (00:32:29): And what we ended up stumbling into is that, "Okay, actually, people are willing to pay more. People want to pay for more inference, because we've crossed this threshold where you can get a very tangible ROI." You know that this is providing a tremendous amount of value to you. So anyways, that was one thing, and the servers were just melting. Anthropic ran out of GPUs for us. Dario emailed me, he was like, "Listen, we don't have anything more to give you." At the times, where we're like, "How do we deal with..." It was just bananas, for weeks. It felt like in 300, when they're surrounded by 10,000 people, and our team is just doing everything. There's 15, 20 people, just doing everything. My chief of staff and I were doing customer support 95% of the day. Anyways. (00:33:22): So yeah, it was a crazy wild time. I mean, it still is. We've had a little bit more time to grow into this. And usually, I mean as a company, to grow into even 20 million ARR, you get a year at least or something, to kind of staff up.

#### 中文翻译:

是的。第一天推文的反响非常好。我们当时想：“哇，这是我们见过的最火爆的发布日反响。”我记得第一天我们增加了大约 6 万美元的 ARR。这太疯狂了，因为我们当时的基数才 60 万，一天就增加了 10%。我记得我们的运维工程师提醒我：“伙计们，今天增加了 6 万，太疯了。”我当时还比较冷静：“是啊，但今天是发布日。”就像初创公司经典的“TechCrunch 效应”高峰一样。我试图平复团队的热情，我说：“这很好，但我们还有很多工作要做。”结果第二天我们又增加了 8 万，然后就一直持续增长。与此同时，我们发布的产品其实只用了 90 天就做出来了。Bolt 是在 90 天内构建的。所以产品里缺少很多东西，非常基础的东西。为了上线，我们确实在一些地方走了捷径。但随着用户大量涌入，人们开始问：“为什么没有移动端适配视图？为什么聊天消息不能……”顺便说一下，我们在没有移动端适配的情况下就达到了 2000 万 ARR。这就像 iPhone 直到 iPhone 5 才有复制粘贴功能一样。对我们来说，当时在手机上看 Bolt 简直糟糕透顶。但就是这样，我们必须应对。我们团队很小，完全没准备好应对不断增长的流量。每天发生的各种问题简直让人抓狂。首先，StackBlitz

以前从未有过超过 9 美元的订阅方案，只有一个 9 美元的价格。发布 Bolt 时我们想：“希望人们喜欢，但 9 美元买不到多少 AI 推理额度。”结果人们在 48 小时内就烧光了 9 美元的额度，然后喊着：“我想买更多，怎么买？为什么不收我的钱？”于是我们在那一周内推出了全新的定价方案，允许升级。这后来成了行业标准，其他竞争对手都抄了这个模式。在 Bolt 上线前，Copilot 等 AI 工具都想学 Netflix 模式，一个价格全包。问题是，如果你这么做，你就必须降低推理成本，因为你预期用户会大量使用，这样你就无法提供那种昂贵的“智能体体验”(agentic experience)。我们无意中发现：“好吧，实际上人们愿意付更多钱，人们愿意为更多推理额度买单，因为我们已经跨过了那个门槛，用户能获得非常切实的投资回报(ROI)。”你知道这为你提供了巨大的价值。总之，那是其中一件事。当时服务器简直在融化，Anthropic 的 GPU 都被我们用光了。他们的 CEO Dario 给我发邮件说：“听着，我们没法再给你更多资源了。”那几周简直乱成一团。感觉就像电影《300 勇士》里被一万人包围的场景，我们 15 到 20 个人的团队在处理所有事情。我和我的幕僚长每天 95% 的时间都在做客户支持。那真是一段疯狂的时光，现在依然如此。通常一家公司增长到 2000 万 ARR 至少需要一年时间来招人。

---

### (00:33:39) Lenny Rachitsky

#### English:

Often, decades.

#### 中文翻译:

通常需要几十年。

---

### (00:33:40) Eric Simons

#### English:

Yeah. So that was as hard, we'd go to people and kind of be like, "What do we do?" And the playbooks we get back are, take six months, or a year, or something. It's like, "This isn't going to work." Which is funny, this is what it's all about. I mean this is, at least for me, that level of intensity, it's challenging. Fun challenges, you know?

#### 中文翻译:

是的。所以这很难，我们去请教别人：“我们该怎么办？”得到的建议通常是“花六个月或一年时间来调整”。但这根本行不通。不过这很有趣，这就是创业的意义所在。至少对我来说，那种强度的挑战非常刺激。

---

### (00:34:08) Lenny Rachitsky

#### English:

Wow, okay. This is just nonstop wild, wild shit. So you mentioned that your team was about 20 people through all of this, you guys are growing at this insane rate, 20 people. How was this possible? What has allowed you to grow this much, this fast, with such a small team? And this 300 visual is interesting, I imagine having these Spartans is a big part of it. Just what has allowed you to do this?

#### 中文翻译:

哇，好吧。这真是一连串疯狂的事情。你提到整个过程中你们团队只有大约 20 人，以这种疯狂的速度增长，却只有 20 人。这怎么可能？是什么让你们在团队如此之小的情况下增长得如此之快？那个《300 勇士》的画面很有趣，我想拥有这些“斯巴达勇士”是关键。到底是什么让你们做到了这一点？

---

**English:**

Yeah, I think a lot of it again, I mean if you kind of look at where a lot of the other folks, like the Cogen types of app space, have really been struggling, a lot of it has been scaling their servers with stuff. And it was kind of like both this overnight success, seven years in the making. All of this stuff, there's no way, if you rewind to year two, there's no way we could have, we would not be at the growth on DAUs, and revenue, or whatever. There's just no way. And so a lot of it is the technology we made, and most importantly, it's been the people. (00:35:19): The people... It's rare to find startups where you have the core group of five, six, seven people, that have been there for five years, plus. That's a pretty rare thing to see in Silicon Valley. It is usually, folks are at a startup for a year or two, they kind of go to another one. You know what I mean? And the problem with the turnover like that is that you can't take really long bets like the one we did. And so we've had, kind of from the get-go, again, this comes back from bootstrapping the previous company. Just having less people, and more context per head. That's just been how we do it, and we feel very strongly about it. (00:36:00): And the reason for that is, one, that you can have high levels of trust with anyone you're talking to, because you know that they have a lot of context. It's not like this person's completely in the dark, in some corner of the company that doesn't... You know what I mean? The second thing, everyone has agency to actually get stuff done, front to back. And there's no political community to get stuff approved by, there's no... So when you look at what happened with Bolt, I mean, we had engineers that were, front to back, were on a call with someone running into an issue, going and fixing it. Cooking up the UI on the spot, and landing this thing. Without involving anyone else on the team. (00:36:41): So I think it was the culmination of just high trust, and people, we all just have enjoyed working together in the past. Maybe that's why, that's the only reason that anyone would ever stay at a company for that long, or whatever. And so those sorts of stressful situations, I think, are make or break. Those are make or break for any team. And so, I think that what's happened is really, it is a direct reflection of the strength and the bonds of the people that are making this thing, and supporting the thing.

**中文翻译:**

是的，我想很大程度上还是归功于技术。如果你看代码生成（Cogen）领域的其他公司，他们很多都在为扩展服务器而苦恼。而 Bolt 就像是“七年磨一剑的一夜成名”。如果回到第二年，我们绝对无法实现现在的日活（DAU）和收入增长，绝对不可能。所以很大一部分原因是我们的技术，但最重要的是人。在初创公司中，很难见到五、六、七个人的核心团队能在一起共事五年以上。在硅谷，人们通常在一家公司待一两年就跳槽了。这种人员流动的代价是你无法进行像我们这样长期的博弈。我们从一开始就坚持少而精，每个人掌握的信息背景（context）更多。这是我们从上一家自筹资金的公司学到的经验。这样做的好处是：第一，你对每个人都有极高的信任，因为你知道他们了解全局，而不是躲在公司某个角落对外界一无所知。第二，每个人都有极强的自主权（agency），能从头到尾把事情办成。没有官僚审批流程。在 Bolt 的案例中，我们的工程师可以直接在电话里帮用户解决问题，当场写好 UI 并上线，不需要牵扯团队里的其他人。所以这是高度信任的结晶，而且我们过去一直很享受在一起工作。这也许是人们愿意在一家公司待这么久的唯一原因。这种高压环境对任何团队来说都是一种考验，要么成就团队，要么摧毁团队。我认为现在发生的一切，正是这群开发和支持产品的人之间强大纽带的直接体现。

---

**(00:37:22) Lenny Rachitsky**

**English:**

Yeah, I think that's such an important point, that you guys have been working together for many years. Most people won't have that benefit. When you're hiring people, when you hire this initial team, is there anything you look for that you think maybe people aren't looking for enough? Anything you prioritize

when you're hiring new folks? Is it this idea that they can do a lot? They can do customer calls, they can do design, they can do engineering?

#### 中文翻译:

是的，我觉得这是一个非常重要的点，你们已经合作多年了。大多数人没有这种优势。当你招聘初始团队时，有没有什么特质是你非常看重，但别人可能关注不够的？你优先考虑什么？是那种“全能型”人才吗？既能接客户电话，又能做设计，还能写代码？

---

### (00:37:42) Eric Simons

#### English:

Yeah, for us, and even if the folks were hiring us, hiring people that don't care about the titles, and they don't care about... It's not like they're... People, of course it's people have a career trajectory, and that sort of thing, but they really are motivated by just working on cool things, and are chucking their ego at the door. And they're there to collectively build something great, not just kind of follow, and be the brilliant jerk. Most of the people that we've hired have been in Europe. We're a fully remote company. My co-founder and I are in the Bay Area. It's funny, back in 2018, we rented an office and stuff, and we were commuting into it. Because we thought we'd hire people here, and like a year into it we were like, "What are we doing? You and I are coming to an office for 10 people, we've hired, the people working for us are in Europe, or across the US." And we have one or two other people we've hired that are in the Bay Area at this point. (00:38:51): But yeah, I think we kind of look for folks that are intrinsically just trying to build great stuff, and are interested. And then the first people that we hired, the reason that we found them is that they were users of StackBlitz. A lot of people, the majority of people we've hired at the company have been people that actually came from our community, basically. So when we want to hire people, we put out a tweet and say, "Hey, we're hiring an engineer," and then we get DMs or whatever. (00:39:22): But yeah, those are the general kind of qualities we look for, though.

#### 中文翻译:

是的，对我们来说，我们招的人不在乎头衔，也不在乎……当然，每个人都有职业规划，但他们真正的动力是做酷的事情，并且能把“自我”（ego）抛在脑后。他们是为了共同打造伟大的东西，而不是做一个自命不凡的混蛋。我们招的大多数人都在欧洲，我们是一家完全远程办公的公司。我和联合创始人在湾区。有趣的是，2018年我们还租了办公室通勤，因为我们以为会在本地招人。一年后我们想：“我们在干嘛？咱俩守着一个能坐10个人的办公室，招的人却都在欧洲或全美各地。”现在我们在湾区也只有一两个员工。我们寻找的是那些发自内心想做大事并对此感兴趣的人。我们招的第一批人其实都是StackBlitz的用户。我们公司的大多数员工基本上都来自我们的社区。所以我们要招人时，发条推特说“招工程师”，然后就会收到私信。这些就是我们看重的一般品质。

---

### (00:39:28) Lenny Rachitsky (Ad Segment: OneSchema)

#### English:

I'm excited to chat with Christina Gilbert, the founder of OneSchema, one of our long-time podcast sponsors. Hi, Christina. (00:39:35): Christina Gilbert: Yes. Thank you for having me on, Lenny. (00:39:37): Lenny Rachitsky: What is the latest with OneSchema? I know you now work with some of my favorite companies like Ramp, Vanta, Scale and Watershed. I heard that you just launched a new product to help product teams import CSVs from especially tricky systems like ERPs? (00:39:53): Christina Gilbert: Yes, so we just launched OneSchema FileFeeds, which allows you to build an integration with any system in 15 minutes, as long as you can export a CSV to an SFTP folder. We see our customers all the time getting stuck with hacks, and workarounds, and the product teams that we work with don't have to turn down

prospects because their systems are too hard to integrate with. We allow our customers to offer thousands of integrations without involving their engineering team at all. (00:40:15): Lenny Rachitsky: I can tell you that if my team had to build integrations like this, how nice would it be to be able to take this off my roadmap, and instead use something like OneSchema. And not just to build it, but also to maintain it forever. (00:40:26): Christina Gilbert: Absolutely, Lenny. We've heard so many horror stories of multi-day outages, from even just a handful of bad records. We are laser focused on integration reliability to help teams end all of those distractions that come up with integrations. We have a built-in validation layer that stops any bad data from entering your system, and OneSchema will notify your team immediately of any data that looks incorrect. (00:40:46): Lenny Rachitsky: I know that importing incorrect data can cause all kinds of pain for your customers, and quickly lose their trust. Christina, thank you for joining us, and if you want to learn more head on over to [oneschema.co](http://oneschema.co), that's [oneschema.co](http://oneschema.co).

#### 中文翻译:

很高兴能与 OneSchema 的创始人 Christina Gilbert 交流，她是我们的长期赞助商。你好，Christina。Christina：谢谢邀请，Lenny。Lenny：OneSchema 最近有什么新进展？我知道你们现在和 Ramp、Vanta、Scale 和 Watershed 等公司合作。听说你们刚推出了一个新产品，帮助产品团队从 ERP 等复杂系统中导入 CSV 文件？Christina：是的，我们刚推出了 OneSchema FileFeeds。只要你能将 CSV 导出到 SFTP 文件夹，就能在 15 分钟内完成与任何系统的集成。我们看到客户经常被各种临时方案困扰，而有了我们，产品团队就不必因为系统难以集成而拒绝潜在客户。我们让客户无需工程团队介入就能提供数千种集成。Lenny：我可以告诉你，如果我的团队要构建这样的集成，能把它从路线图中拿掉并使用 OneSchema 这样的工具该有多好，不仅是构建，还有永久的维护。Christina：没错。我们听过太多因为几条错误记录导致系统宕机数日的惨剧。我们专注于集成的可靠性，帮助团队消除这些干扰。我们有内置的验证层，防止错误数据进入系统，并会立即通知团队。Lenny：我知道导入错误数据会给客户带来巨大痛苦并失去信任。Christina，谢谢你的加入。想了解更多请访问 [oneschema.co](http://oneschema.co)。

---

## (00:41:01) Lenny Rachitsky

#### English:

I want to ask a couple more questions about Bolt, and then I want to zoom out, and talk about where things are heading in the future. Let's talk about prioritization. I imagine you guys are just barraged with, as you described, after you launched, you're just barraged with requests. Like you said, there's a million monthly active users. I can't even imagine the feature requests you guys are getting, plus all the stuff you know want to build. Just how do you go about deciding what to prioritize, and what to actually build?

#### 中文翻译:

我想再问几个关于 Bolt 的问题，然后我们放大视角，聊聊未来的走向。谈谈优先级（prioritization）吧。我能想象你们在发布后被各种请求淹没了。就像你说的，有 100 万月活用户，我无法想象你们收到了多少功能请求，再加上你们自己想做的东西。你们是如何决定优先处理什么，以及实际构建什么的？

---

## (00:41:28) Eric Simons

#### English:

There's a lot of things that you just don't even know are possible to do, and so, people aren't going to be necessarily explicitly asking for them. And so there's been kind of a couple of these, where we use our gut instinct on like, "Hey, no one's asking for this, in meaningful numbers at least. But we think this is going to be a big deal." (00:41:47): Best example was last week with native mobile app support. That's, by reception, the biggest thing we've ever launched. And it was something that, even internally at the company, some folks were like, "This, I don't know. People are yelling about these other things." And it is,

it's always this balance of how much are we just triaging various things, versus that new capabilities, but it was like, "This strikes me as an important one," where we put some chips into the middle of the table on. And had it dead right. It's just this mind-blowing experience, and now, there's just thousands of mobile apps being created a day, that weren't before. And how does that change things? I mean, now there's small businesses that, they would've never made an iPhone app before. It made no sense. It's super expensive. Now, that's not the case. (00:42:43): So there's kind of these things where it's like, "Hey, we should go and take bets here." But there's kind of this, I think the best analogy would be like, it's kind of like working at a restaurant, being like a chef. There's some amount of, there's feedback from the customers of, "This thing didn't taste good." And then there's like, "Hey, we've been cooking something interesting, and this tastes... I don't know. This, I think you're going to like this. I think this is a killer dish." And so you kind of have to balance those things. (00:43:15): And I think it's actually, largely, a function of just years of experience doing it. I think if you kind of rewound 10 years ago, I would have had really no, I wouldn't have had just the years of getting my butt kicked by the free market, to have cultivated a sense of this stuff. You kind of have to build your own gun and stick for it, I guess is the best way of putting it.

#### 中文翻译:

有很多事情用户甚至不知道是可以做到的，所以他们不一定会明确要求。在这种情况下，我们会凭直觉判断：“嘿，虽然还没多少人要求这个，但我们认为这会是个大动作。”最好的例子就是上周推出的原生移动应用支持。从反响来看，这是我们发布过的最重大的功能。即使在公司内部，也有人犹豫：“我不确定，大家都在喊着要别的东西。”这总是在处理琐碎反馈和开发新能力之间寻找平衡。但我当时觉得：“这很重要。”于是我们押注了它，结果证明完全正确。这是一种令人震撼的体验，现在每天都有成千上万个以前不存在的移动应用被创建出来。这改变了什么？这意味着以前从未想过做 iPhone 应用的小企业（因为太贵太难），现在可以做了。所以有些时候你需要去“下注”。最好的类比是当厨师。一方面有顾客反馈“这个不好吃”，另一方面你会说“嘿，我们研发了个新菜，我觉得你会喜欢的，这是招牌菜”。你必须平衡这两者。我认为这很大程度上取决于多年的经验。如果回到 10 年前，我可能还没有这种被自由市场“毒打”多年后培养出来的直觉。你必须建立自己的判断准则。

---

## (00:43:39) Lenny Rachitsky

#### English:

To unpack this a little bit further, do you have a cadence you guys work through to decide what to build, and ship? Do you have a weekly meeting every week? Because I know the answer is probably, really, "It's just chaos constantly, and fires we're putting out constantly." I know that's a lot of it, but is there some kind of process that you guys have for deciding what to build, and how to share it, and just work with the team?

#### 中文翻译:

进一步拆解一下，你们决定构建和发布什么时有固定的节奏吗？每周开会吗？因为我知道答案很可能是“一直处于混乱中，一直在救火”。但我还是想知道，你们有没有某种流程来决定做什么、如何分享并与团队协作？

---

## (00:44:02) Eric Simons

#### English:

We all meet every day. Pretty much the entire team gets on a call, and we just kind of front to back- (00:44:07): Lenny Rachitsky: You meet, like a Zoom? (00:44:08): Eric Simons: Yeah, every day at 8:00 AM Pacific, we're on a Zoom for at least an hour- (00:44:12): Lenny Rachitsky: Every day? The whole company? (00:44:13): Eric Simons: Pretty much the entire company. Yeah. (00:44:14): Lenny Rachitsky: Wow. For an hour. Okay. (00:44:16): Eric Simons: Yeah. And we just go over everything and I think we're

going to probably start, as the team's kind of growing, we're going to start splintering off into different syncs, or whatever. But the thing about just having everyone in the same room every day is that, a lot of people will complain that it's... On Twitter, you'll see people say, "Oh, it's the most expensive use of everyone's time," but it's like, "Yep. But there's 0% fidelity loss in that. Everything, every day, is being audited front to back, and being discussed front to back." (00:44:47): So when you're in these times of just extreme growth, you want as close to 0% loss, on communications. And so that's how we've been doing it, especially since Bolt went online, and I think it was the week after Bolt went online we were like, "Every day, until we're through this or whatever, we're all getting on a phone call every day, and we're front to back doing this." And again, another reason why more context and less heads, every person at the company is aware of everything else going on at the company. So people can independently be making decisions that are generally, by default, more often correct than not.

**中文翻译:**

我们每天都见面。几乎全团队都会参加视频会议，我们从头到尾梳理……Lenny：是用 Zoom 吗？Eric：是的，太平洋时间每天早上 8 点，至少一小时。Lenny：每天？全公司？Eric：几乎全公司，是的。Lenny：哇，一小时，好吧。Eric：是的。我们会过一遍所有事情。随着团队扩大，我们可能会开始拆分成不同的同步会议。但每天让所有人都在同一个“房间”里的好处是，虽然推特上很多人会抱怨这是“对所有人时间最昂贵的浪费”，但事实是，这样沟通的保真度损失为 0%。每天的所有事情都会被从头到尾地审计和讨论。在极速增长时期，你希望沟通损失接近 0%。这就是我们的做法，尤其是 Bolt 上线后的那一周，我们决定每天都开会，直到渡过难关。这又是“信息背景多、人头少”的好处，公司里的每个人都了解其他人在做什么。这样人们就能独立做出决策，而且默认情况下，这些决策通常是正确的。

---

### (00:45:33) Lenny Rachitsky

**English:**

That is so interesting. I've never heard that before. Especially for company growing, that is like yours. That is super interesting, that that's what you do.

**中文翻译:**

这太有意思了，我从未听说过这种做法，尤其是像你们这样高速增长的公司。这真的很特别。

---

### (00:45:42) Eric Simons

**English:**

I don't think we're going to do that forever. (00:45:44): Lenny Rachitsky: Yeah, yeah. Of course. (00:45:45): Eric Simons: But, yeah.

**中文翻译:**

我想我们不会永远这么做。Lenny：当然。Eric：但目前是这样。

---

### (00:46:01) Lenny Rachitsky

**English:**

Where do you, so say you talk about stuff, then where do you put stuff? Where do you put your roadmap? Where do you plan? Just, what tools are kind of in the stack of the company's toolset?

**中文翻译:**

## (00:46:01) Eric Simons

### English:

Yeah, on the engineering side, we use Linear heavily. On kind of product roadmapping, we're using Notion, and kind of making PRD type stuff in Notion. And we use Figma for design. No, actually, we use Bolt for a lot of design and prototyping at this point, as you can imagine. But yeah, I think that the tooling is nothing crazy. There's nothing crazy sophisticated. I think we'll be investing a lot more, and especially as you start splintering people out of being on the same call every day, so that's where this stuff really starts to matter. Because you don't have a time where you're able to dynamically catch things that weren't going to be brought up.

### 中文翻译:

在工程方面，我们大量使用 Linear。在产品路线图方面，我们用 Notion，并在上面写 PRD（产品需求文档）。设计用 Figma。哦不对，实际上我们现在很多设计和原型制作都是直接用 Bolt 完成的，你可以想象得到。工具方面没什么特别复杂的。我想随着团队不再每天全员开会，我们会在这方面投入更多，因为到那时，工具的重要性就会凸显出来，因为你不再能动态地捕捉到那些没被提出来的问题。

---

## (00:47:01) Lenny Rachitsky

### English:

I love that you guys use PRDs. I love that you even used that term. There's a lot of talk of just like, "Oh, we got Bolt now we got all these tools, we don't need PRDs. We're just going to create a prototype immediately, and that's it." Talk about just why you still find that useful, and just what you put into your PRD, whatever that is for you.

### 中文翻译:

我很高兴你们还在用 PRD，甚至还用了这个词。现在很多人在说：“哦，我们有了 Bolt 这种工具，不需要 PRD 了，直接做个原型就行了。”聊聊为什么你觉得它依然有用，以及你的 PRD 里通常包含什么？

---

## (00:47:18) Eric Simons

### English:

Unless there's something that's very sophisticated that we're working on, we tend to keep them pretty light. I like to just have the minimal amount of context possible, that just ensures everyone's on the same page and that the key outcomes for whatever feature that we're working on, are going to be present when we get there. Because the things that, when these arguments get really beefy, you're looking at it at, "God, there's so much stuff to decipher." The problem is, a lot of people are going to gloss over it when it gets kicked to development, or design, or whatever. It's just going to start snowballing into a lot of stuff. It's just better to keep it as simple as you possibly can. At least, that's our approach to the thing. And often it's some of these things are like, "Here's a link to a Bolt." (00:47:46): Lenny Rachitsky: "And here's what it might look like." (00:47:51): Eric Simons: Yeah. And not just look like, "Here's kind of a working demo of what it will effectively feel like." And then, be. Because that just, if a picture is worth a thousand words, a live actual demo is worth millions. You can feel it. It's real. And that's what we're seeing, a lot of the businesses that are adopting Bolt now, that's the use case that they're using this for. Is high fidelity prototyping, because it's now faster to make real prototypes using Bolt. Before, it was too expensive. The

idea of, "And let's prototype it, the engineers' code a proto..." It's like, that, it would take forever. It would be expensive. And now it's faster to do this with Bolt, in code, and have a real working software product, than dragging around frames and Figma to actually make a static version of it.

#### 中文翻译:

除非是非常复杂的东西，否则我们的 PRD 都很简练。我喜欢只保留最核心的信息背景，确保大家步调一致，明确功能的关键产出。因为如果文档太厚，大家在开发或设计时往往忽略掉。事情会像滚雪球一样变得复杂。所以最好保持简单。我们的做法通常是：“这是 Bolt 的链接。” Lenny：“这就是它可能的样子。” Eric：是的，不仅是样子，而是“这是一个可运行的演示，告诉你实际感觉如何”。如果说一张图胜过千言万语，那么一个活生生的演示就胜过百万言语。你能感受到它，它是真实的。我们看到很多采用 Bolt 的企业正是为了这个用途：高保真原型制作。因为现在用 Bolt 制作真实原型的速度更快。以前这太贵了，让工程师写个原型代码要花很长时间。现在用 Bolt 写代码做原型，比在 Figma 里拖拽框架做一个静态版本还要快。

---

### (00:49:08) Lenny Rachitsky

#### English:

So let's actually talk about that. Just how far have companies gotten with Bolt? Prototypes is where everyone's kind of imagining these tools are at. I know that the goal isn't just to make prototypes, it's to build full scale. I imagine, long-term, Salesforce, Atlassian style companies, at scale. What are some examples of products people have built with Bolt that maybe would surprise people, in just how far they've gotten?

#### 中文翻译:

那我们聊聊这个。公司用 Bolt 到底做到了什么程度？大家通常认为这类工具只能做原型。但我知道目标不仅仅是原型，而是构建完整的规模化应用。我能想象长远来看，会出现 Salesforce 或 Atlassian 这种规模的公司。有哪些人们用 Bolt 构建的产品例子会让大家感到惊讶？

---

### (00:49:30) Eric Simons

#### English:

Yeah, I mean, especially... When you're starting greenfield stuff, you can use Bolt to build... you know. Like Salesforce, as an example. One of the first people that signed up for Bolt was this guy named Paul, and he's an entrepreneur, and doesn't know how to code. Built a CRM in three weeks, that has AI built into it and Stripe for billing, et cetera. He had gotten a quote from an agency for this, it was going to be 30 grand, and take six months. He had done in three weeks, and I think he spent 300 bucks on Bolt for the thing. So it's like, this is... And he's making money off of this. This is his start. Right? (00:49:45): Lenny Rachitsky: Okay, so he built this, and he's selling it. People are paying to use it. (00:49:48): Eric Simons: Yeah. Yeah. (00:49:48): Lenny Rachitsky: Wow. (00:49:50): Eric Simons: And there's many such cases of this. If you're looking at greenfield projects, 100%. Today with this current state of frontier models, you could absolutely build production grade software. You're not going to get a zero shot, but you're going to spend a couple of days, weeks, whatever. But the cost reduction there, 30 grand versus \$300. It's 99% cheaper. Six months versus three weeks. I mean, it's like order of magnitude sort of fashioned delivery on the thing. And those numbers have helped, for the people that we talk to, that are building these full stack apps. People, they go to Upwork, work, they get a quote for five grand. They have it within 50 bucks. It's just nuts, what you're able to do with this thing. (00:50:35): And so I think on the flip side, a lot of the existing companies, there are very legitimate use cases where things are greenfield, spun up. A good example is public websites. Marketing pages, landing pages, whatever have you. Folks are adopting Bolt to just power those instead of using Webflow, for example. Because it's like, this is simpler to use than

Webflow. And it integrates with the existing design system of the company, and et cetera. And the marketers can update it without knowing how to code, whatever. But then for product development teams, this is most commonly for, again, existing software businesses. They're using this to just accelerate the product development process, and in a way where it's not just like a Greenfield wholesale, "Hey, we're building the entire thing in Bolt," or whatever.

**中文翻译:**

是的，尤其是当你从零开始（greenfield）做新项目时。比如 Salesforce。Bolt 最早的一批用户里有个叫 Paul 的创业者，他不会写代码。他在三周内做了一个 CRM（客户关系管理系统），内置了 AI 功能，还集成了 Stripe 支付。他之前找外包公司报价，对方要 3 万美元，工期 6 个月。他自己用三周做完了，在 Bolt 上只花了 300 美元。而且他现在已经靠这个赚钱了，这就是他的初创公司。Lenny：所以他做出来并开始卖了，有人付钱用它。Eric：是的。这样的案例很多。对于新项目，百分之百可行。以目前的尖端模型水平，你完全可以构建生产级软件。虽然不是“一语即成”（zero-shot），但你花几天或几周就能搞定。成本从 3 万降到 300，便宜了 99%。时间从 6 个月缩短到 3 周，这是数量级的提升。对于那些在 Upwork 上询价 5000 美元的人来说，现在 50 美元就能搞定，这简直疯了。另一方面，对于现有公司，也有很多合理场景。比如公开网站、营销页面、落地页。人们开始用 Bolt 而不是 Webflow，因为它比 Webflow 更简单，还能集成公司现有的设计系统，营销人员不需要懂代码就能更新。对于产品开发团队，他们用它来加速开发流程，而不一定是把整个老系统搬到 Bolt 上。

---

## (00:51:20) Lenny Rachitsky

**English:**

Can Bolt integrate with your existing code base, or not yet?

**中文翻译:**

Bolt 现在能集成现有的代码库吗？还是还没实现？

---

## (00:51:23) Eric Simons

**English:**

So yeah, we can actually open up repos in Bolt. You can go and use Bolt on your code base. It kind of depends on your setup. And we do have companies, again, that have marketing sites they're using this on, or their admin panel or whatever. And I think it's going to be a use case that we see a lot more people orienting towards. These LLMs are not great, depending on how big your application is, though. These things are not quite there, where if you have something that's a thousand files or something, or more, where you're going to be able to have a really reliable, super reliable experience per se. Within a year, we'll chat a year from now, I suspect the answer is going to be different. So it kind of depends on the size of the app, the scale of the app, and if it's too big, you're looking at the prototyping, just pure acceleration of product development. And if it's not, then you can just do it entirely from Bolt.

**中文翻译:**

可以，我们实际上可以在 Bolt 中打开代码仓库。你可以在你的代码库上使用 Bolt，这取决于你的配置。有些公司在营销网站或管理后台上使用它。我认为我们会看到更多人转向这个方向。不过，大语言模型（LLM）的表现取决于应用的大小。如果你的项目有上千个文件，目前的体验还不够极致可靠。一年后我们再聊，我猜答案会不一样。所以这取决于应用的规模。如果太大，它更适合做原型和加速开发；如果规模适中，你完全可以在 Bolt 里搞定。

---

## (00:52:24) Lenny Rachitsky

## English:

So this is useful. So what would you say are the major limitations of Bolt today, where people should just know, "Okay, it's not going to get you here yet. Maybe in the future it will." So it sounds like, if you have a really large existing code base, probably not the best tool yet. What else should people know?

## 中文翻译:

这很有用。那么你会说目前 Bolt 的主要局限是什么？大家应该知道哪些地方它目前还做不到。听起来，如果你有一个非常庞大的现有代码库，它可能还不是最佳工具。还有别的吗？

---

## (00:52:39) Eric Simons

### English:

I would say that's probably the main one, because I think if you have a large existing code base, you're going to need something like Cursor. And you're going to need to be a developer, meaningfully, to be editing that stuff. I think outside of that, there's a, just like using any other productivity tool, like Photoshop or Figma, or like a DSLR or whatever. There's some level of education, and using the tool, and learning how to use it, that's required to really unlock a lot of the maximum capabilities of the thing. (00:53:10): And the people that we see that are most successful with Bolt, outside of developers, the people we see that are most successful are people that are amazing PMs, for example. Because these are people that understand enough about how the technology works, typically, and their job is to direct developers on how to go and improve the product. And go and look into how to actually spec this thing out in a way that's executable, without lossiness in the communication. And when you think about, "Okay, how would you best interact with an AI developer agent?" It's basically that. You really want to be good at defining scope, and helping it go and debug various things, or whatever have you. There's a huge overlap of the skill set of being a rock star PM, and being really good at using, frankly, any of these text to apps or Cogen tools.

## 中文翻译:

我想那是主要的局限。如果你有庞大的代码库，你可能需要 Cursor 这种工具，而且你必须是一个真正的开发者才能修改它。除此之外，就像使用 Photoshop、Figma 或单反相机一样，你需要一定的学习成本来解锁工具的最大潜力。我们看到除了开发者之外，在 Bolt 上最成功的人通常是优秀的产品经理（PM）。因为这些人通常足够了解技术原理，他们的工作就是指导开发者如何改进产品，并制定出可执行且无损沟通的规格说明。当你思考“如何最好地与 AI 开发智能体互动”时，本质上就是这些。你需要擅长定义范围（scope），并协助它调试。一个顶尖 PM 的技能集与使用这些“文本生成应用”或代码生成工具所需的技能高度重合。

---

## (00:54:08) Lenny Rachitsky

### English:

I love that you made that point. That's exactly the point I've been trying to make, I have a newsletter post about this. Because when all these tools came out, there was so many people saying, "Okay, PMs are dead. We don't need them anymore. We can just build things so quickly and easily, what's the point?" But I completely see the world the way you see it. The hard part now is, now it's easy to build the thing. Now it's, "What the hell should we build? Can we clearly articulate what it is we want to build?" And then, "Can we just have the taste to know, is this right, is this correct? Is this good? Is this going to solve the problem?" And then it's like, grow it, which is something also PMs think about. So, I completely agree. Basically it feels like PMs are, and a lot of PMs listen to this, so they'll love hearing this. To me, it feels like PMs are the best positioned role to thrive in this world.

## 中文翻译:

我太喜欢你这个观点了。这正是我一直想表达的，我还为此写过一篇 Newsletter。因为当这些工具出现时，很多人说：“PM 完蛋了，我们不再需要他们了，既然构建东西这么快这么简单，PM 还有什么意义？”但我的看法和你完全一致。现在的难点在于，构建变得容易了，但“我们到底该构建什么？”“我们能否清晰地表达出想构建的东西？”以及“我们是否有足够的品味去判断这是否正确、是否好用、是否解决了问题？”还有如何让它增长，这些都是 PM 思考的事情。所以我完全同意，PM 似乎是这个新世界里最能大放异彩的角色。

---

## (00:54:54) Eric Simons

### English:

Zero question. I mean that was, as Bolt was growing and we were like... Because we were a developer product before this, and so we expected the audience to be 100% developers that were using this. And we just kept seeing more and more and more people that were not developers using it, to the point where it's like, 67% of our users are not developers, at this point. And when I started talking to these folks, at first I was just weird, or whatever. It was like, "Well, what's going on here?" But then it just kind of clicked as like, "Oh, well, this is going to change everything. The entire software world order is going to get rewritten, here." (00:55:30): Because the way that companies are organized to build software today, totally going to change. The idea that again, PMs are the people that really understand, to the pixel level, what matters into making a great product experience. And often they're having... And listen, I'm a developer, myself. They have to go and harangue the developers to get things to be how they really ought to be, to the smallest levels. And now, how this is going to work, if you fast-forward one, two, five years, whatever. PMs, they're going to be "writing code", quote, unquote, instead of just writing a JIRA ticket and waiting for a developer to do it. The developers are going to be able to work on intellectually challenging tasks that LLMs are not well suited for, and still being augmented by LLMs to do it. But PMs are going to be able to go in and just make the changes themselves. (00:56:24): And what blew my mind is, it's not priced in, to any of these companies out there. And it's not reflected in the org charts of all the software companies in the world right now. That is going to completely change. The winners, at least, their org charts are going to completely change, and how they approach building products and shipping products. Completely. And it's starting, this is the beginning.

### 中文翻译:

毫无疑问。随着 Bolt 的增长，我们发现……因为我们以前做的是开发者产品，所以预期用户 100% 是开发者。但我们不断看到越来越多的非开发者在使用它，现在 67% 的用户都不是开发者。刚开始和这些人聊时我觉得很奇怪，但后来我突然明白了：“哦，这会改变一切。整个软件世界的秩序将被改写。”因为今天公司构建软件的组织方式将彻底改变。PM 是那些真正理解像素级产品体验的人。以前他们（我也是开发者）必须不断催促开发者，才能把细节磨到理想状态。快进到一两五年后，PM 将会“写代码”（打引号的），而不是写个 JIRA 工单然后等开发者去实现。开发者将专注于 LLM 不擅长的智力挑战任务，并由 LLM 辅助。而 PM 可以直接进去自己做修改。令我惊讶的是，这种变化还没有反映在目前任何公司的估值或组织架构图中。但这将彻底改变。至少赢家的组织架构和打造、发布产品的方式将完全不同。这仅仅是个开始。

---

## (00:56:50) Lenny Rachitsky

### English:

I want to follow that thread, but first of all, I want to also add, and correct me if you disagree with this. I think when we talk about PMs, that also applies to founders, like product thinking founders- (00:56:58): Eric Simons: One hundred percent. (00:56:59): Lenny Rachitsky: ... very similar. And then, I think it's also important to note, if you also have engineering skills and design skills, you will be at an advantage. That

only helps you. But if you're looking at this triangle of the triad of product, engineering, PM, it feels like the PME skills are the ones that will be most important and valuable. Although it'd be great if you can also be in code, and if you could also design really well.

#### 中文翻译:

我想顺着这个话题聊，但首先我想补充一点（如果不对请纠正我）：当我们谈论 PM 时，这也适用于创始人，即具有产品思维的创始人。Eric：百分之百。Lenny：非常相似。而且我认为很重要的一点是，如果你同时具备工程和设计技能，你会有巨大优势。但在产品、工程、PM 这个铁三角中，PM 的技能似乎变得最重要。尽管如果你能写代码、懂设计会更好。

---

### (00:57:23) Eric Simons

#### English:

Absolutely. And to me, it's the most exciting mix. I think PMs, designers and entrepreneurs that are non-technical, that's the most exciting thing to me, just because it's a brand new market that's being unlocked here. For the first time ever, these folks can directly code and build the product, themselves. Their vision, directly into the software itself. That's going to change everything. That is changing everything.

#### 中文翻译:

绝对是。对我来说，这是最令人兴奋的组合。非技术背景的 PM、设计师和创业者，这是最让我激动的，因为一个全新的市场正在被开启。有史以来第一次，这些人可以直接“编写”并构建产品，将他们的愿景直接转化为软件。这正在改变一切。

---

### (00:57:57) Lenny Rachitsky

#### English:

So when you talk about how org charts are going to change, what are you imagining there? Is it just fewer engineers, mostly, or what do the future org charts look like?

#### 中文翻译:

当你谈到组织架构将如何改变时，你脑海中是什么样子的？主要是工程师变少了吗？未来的组织架构图长什么样？

---

### (00:57:57) Eric Simons

#### English:

Good question. And I bet you there's going to be some Gartner analysis someday, years from now or whatever that's like, "Here's how the best," some term is applied to how the best companies are organizing. But yeah, I think that we're going to see developers probably being pulled off of a lot of the, generally speaking, pulled off of a lot of user interface type work. I would imagine. Except for the most complicated of those things. And you're going to see designers and PMs really, really leading the charge, and being responsible for crafting those experiences. And perhaps having a developer attached to be reviewing the code and making sure the guiding, the code that they're writing, reviewing those pull requests and et cetera. And I think maybe even the engineers are... Like you pointed out, having engineering skills is not going to hurt you. It's going to make you way more effective. (00:58:58): But I do think there's going to be, the leverage that the front engineer is going to have is, it is now insane, it's only

going to get more so. And so I could see there just being fewer front engineers attached to, I'm seeing more product and design folks, with one or two engineers or something. And really having a larger matching of pods like that. Something like that strikes me as probably how this is going to start trending towards.

#### 中文翻译:

好问题。我敢打赌，几年后 Gartner 肯定会出一份分析报告，给这种新型组织方式起个名。但我认为，开发者可能会从大量的用户界面（UI）工作中抽离出来，除非是极其复杂的 UI。你会看到设计师和 PM 真正主导这些体验的打造。也许会配备一名开发者来负责代码审查（Code Review），确保代码质量并提供指导。就像你说的，懂工程只会让你更高效。但我认为前端工程师的杠杆率现在已经高得惊人，而且还会继续提高。所以我预见未来每个小组（pod）里前端工程师会变少，而产品和设计人员比例增加，配上一两个工程师。这种模式可能会成为趋势。

---

### (00:59:34) Lenny Rachitsky

#### English:

This touches on, we had a researcher from OpenAI in the podcast. She actually started her career, she worked at Anthropic first, as a front-end engineer. And said that once she saw what Clyde could do, for front-end engineering, she's like, "I need to move to a different function." And so she moved into research, because she saw that role disappearing, potentially. And that's exactly what you're saying.

#### 中文翻译:

这让我想起我们播客请过的一位 OpenAI 研究员。她职业生涯初期在 Anthropic 做前端工程师。她说当她看到 Claude 在前端工程方面的表现后，她心想：“我得换个职能了。”于是她转到了研究岗，因为她预见到那个角色可能会消失。这正印证了你的观点。

---

### (00:59:57) Eric Simons

#### English:

Yep.

#### 中文翻译:

没错。

---

### (01:00:19) Lenny Rachitsky

#### English:

So let me ask you this, I don't know if you have a clear thesis on this yet, but say, you just had a kid. Say your kid is, in the future, starting school. Let's say your kid was starting college soon. Do you have thoughts on just what skills slash areas you think they should go into, versus avoid, that maybe are popular now and are going to be less popular?

#### 中文翻译:

那我问你，我不知道你是否已经有了明确的论点，但假设你刚有了孩子，假设你的孩子未来要上学，或者快上大学了。对于他们应该学习哪些技能/领域，以及避开哪些现在流行但未来可能贬值的领域，你有什么想法吗？

---

**English:**

Understanding how to leverage these AI tools is key. I wouldn't necessarily, I think maybe getting a basic understanding of how programming works, et cetera, is great. (01:00:31): Lenny Rachitsky: It's like technical foundations, just understanding how systems work, how coding works. (01:00:41): Eric Simons: Exactly. But it doesn't have to be, because I think back to if Bolt existed, like Albert and I say this to each other all the time. Since the get-go, StackBlitz, we've been building the thing that we wish we had when we were 13. And heck, for everything we built since then. And especially with Bolt. I mean, I don't know if I would've gone as deep as I did on learning how to code, and being an engineer, if that had been around then. The whole reason we got into it is we had ideas for products, and businesses, that we wanted to build. And coding was just a necessary requirement in order to do that. (01:01:21): And that said, I think people need to follow their intrinsic interests. If folks are really interested in really getting in the nitty-gritty of how computers work, and program leaders, or spark and compilers, or whatever, go for it. I think that stuff is still going to be relevant. I don't know if we're going to really have, we'll see, but to the degree that there's AGI where it's like, we don't have to think about anything ever again. (01:01:47): Lenny Rachitsky: Yeah, that's always the answer here. Do we sell anything? (01:01:50): Eric Simons: Yeah, it's like if we're at that point, it's kind of... I don't know, I'm not sure. But I think from, at least what I feel like seems like the next at least five years of what we're looking at, I think people are still going to, there's still going to be places to specialize, and really go deep. But I think you want to go into it with the idea, not like, "I'm going to go and learn computer science because I'm going to get a job for sure out of it." (01:02:18): I just think that's generally not a good... This is like, my co-founder and I, we didn't go to college. My co-founder dropped out of college after a semester or something, but I didn't go, because I was like, "We're coding," we were doing contracting at the time, making money. And it was like, "This is a lot of," you know, it would've been like a hundred grand of debt by the end of the thing, just for four years of in-state tuition, at U of I. 120 grand, I think, at that time. (01:02:49): And lo and behold, I mean there's a huge issue with this. Where people are kind of... There was a prevailing thought by society that going to college in the early 2010s or late two 2000s, that you're going to get a job on the other side. That's going to be high paying. And that just has not been the case for a lot of people. And I think that's just going to continue to be the case. But again, not to deter people from doing it, but you have to go into it being like, "I for sure, this is what I want, and I want to go and be the best that I can possibly be at this thing." You know what I mean?

**中文翻译:**

理解如何利用这些 AI 工具是关键。我不一定觉得要钻得极深，但掌握编程的基本原理是很棒的。Lenny：就像技术基础，理解系统如何运作，代码如何运作。Eric：没错。但不一定要以此为生。我和 Albert 常说，StackBlitz 一直在做我们 13 岁时梦寐以求的东西。如果当时就有 Bolt，我不知道我还会不会在学习编程和成为工程师上钻研得那么深。我们进入这个领域的初衷是有产品和业务想法，而编程只是实现它的必要手段。话虽如此，我认为人们应该追随自己的内在兴趣。如果有人真的对计算机底层原理、编译器等感兴趣，那就去做。这些东西依然会有相关性。除非 AGI（通用人工智能）真的发展到让我们再也不需要思考任何事情的地步。Lenny：那是终极答案。Eric：如果到了那一步，我也不确定。但至少在未来五年，我认为依然有专业化和深耕的空间。但你进入这个领域不应该是为了“学计算机肯定能找到好工作”。我和联合创始人就没上大学。他上了一个学期就退学了，我压根没去，因为当时我们已经在写代码赚钱了。当时如果去上学，毕业时会背负 10 万美元的债务。现在社会上有一个大问题：2010 年代初人们普遍认为上大学就能拿高薪，但事实并非如此。我认为这种情况会持续。所以不是阻止大家上学，而是你必须抱着“这确实是我想要的，我要做到最好”的心态去学。

## English:

I like the transfer to your kid is going to be like, "Don't even go to college," potentially.

## 中文翻译:

我喜欢这个转折，你对你孩子的建议可能是：“甚至可以不用去上大学。”

---

## (01:04:03) Eric Simons

### English:

Only if they want to. I think at 18, it's a huge ask. I mean, it's a huge ask, not even at 18. It's like at 17, because you could go apply for colleges. It's just such a huge... Like, a six figure debt commitment to someone who's making \$0, or negative dollars, and that young. Unless you really have conviction, it costs nothing to go and explore and learn for free, online.

### 中文翻译:

除非他们想去。我认为在 18 岁（甚至 17 岁申请大学时）就让一个零收入的年轻人背负六位数的债务，这是一个巨大的要求。除非你真的有坚定的信念，否则在网上免费探索和学习是不花钱的。

---

## (01:04:11) Lenny Rachitsky

### English:

I want to come back to the skills that you think are going to be most important, and let me try to mirror back a few things you said, that I very much agree with. (01:04:11): So it feels like if you want to be successful in the world where AI can build things for you, more and more, what I'm hearing is get good at figuring out what people need and want, what problems they need solved. Get good at articulating it really well to the AI tools. And there's this talk, "You don't need to be a great prompt engineer. You don't need to work on prompting," but it feels like it's more and more important, because you tell it something and it goes off and builds a thing. If you're clear about it, it'll save you a lot of time. (01:04:42): So it's be good at figuring out problems people need solved, figure out how to articulate that problem well, and ask for a clear solution. Figure out how to grow the thing, feels like that's still going to be a need, because Bolt's not going to go and find every... I could see that still running paid ads, and stuff like that. But it feels like that's going to be ongoing need. (01:05:03): And then I feel like there's this kind of unstuck step, like helping AI get unstuck, and it feels like that's where maybe engineering skills will come in more and more. Thoughts on just that skill?

### 中文翻译:

我想回到你认为最重要的技能上。让我试着总结一下你说的几点，我非常同意。在 AI 越来越多地为你构建东西的世界里，成功的关键是：擅长发现人们的需求和想要解决的问题；擅长向 AI 工具清晰地表达这些需求。虽然有人说“不需要成为提示词工程师”，但我觉得表达能力越来越重要，因为如果你表达清晰，能节省大量时间。所以，技能包括：发现问题、清晰表达问题并索要解决方案、以及如何让产品增长（因为 Bolt 不会自动帮你找客户或投广告）。最后，还有一个“帮 AI 脱困”的步骤，这可能是工程技能发挥作用的地方。你对这项技能怎么看？

---

## (01:05:13) Eric Simons

### English:

Oh, totally. Yeah. So we actually, two weeks ago I think, we announced this program called Bolt Builders. And it's basically the genius bar at the Apple Store, where as folks are building on Bolt that are not developers, they'll run into some nook or cranny where the AI just cannot figure it out, or whatever. And I think that's just going to continue to be the case for the time to come. That's our position, and that's why we spun up this program. (01:05:38): Lenny Rachitsky: And these are humans, that help you out. (01:05:41): Eric Simons: These are humans. And people that we're certifying. And so, in Bolt in the coming weeks or whatever, there's going to be a button where you can just say, "Hey, connect me with a certified expert." And you can chat with them live, and they'll help you get unstuck, and you pay I don't know, 50 bucks an hour. Whatever it is. And then you get unjammed and you keep prompting. And again, I just think this stuff is, it just all seems like gravy to me. Engineers get to focus on difficult challenges, not like cookie cutter, "Let's make another CRUD app," stuff. They get to, debugging is challenging, and fun. And going and working on intellectually stimulating tasks, and all this stuff that's just copy pasta, over and over, all this, error app. It's just like, let the AI do all that kind of crap.

#### 中文翻译:

噢，完全同意。实际上，大约两周前我们宣布了一个叫“Bolt Builders”的项目。它基本上就像苹果商店的“天才吧”(Genius Bar)。当非开发者在 Bolt 上构建时，可能会遇到 AI 无法解决的死角。我认为这种情况在未来会一直存在。这就是我们的立场，也是我们启动这个项目的原因。Lenny: 这些是帮助你的人类。Eric: 是的，是人类，而且是我们认证的专家。在接下来的几周里，Bolt 会出现一个按钮，你可以说“帮我联系认证专家”。你可以和他们实时聊天，他们帮你脱困，你付一定的时薪（比如 50 美元）。解决后你继续用提示词构建。我觉得这太棒了。工程师可以专注于真正的难题，而不是重复造轮子做那种简单的增删改查 (CRUD) 应用。调试是有挑战且有趣的，去做那些有智力启发的工作，至于那些重复性的、无聊的代码，让 AI 去处理那些垃圾活儿吧。

---

## (01:06:58) Lenny Rachitsky

#### English:

This is a potentially new job, for now at least, just unstuck the AI. Which I think over time, it'll get better and better, and we maybe won't need these people. But I love that it's now AI first, and person second. Versus person building the thing, and then AI. Like when Copilot launched, it was just like, "Cool, here's a little suggestion for this function," and now it's flipped. "Here's everything." And then, "Oh, I don't know what to do here. Help us here." And then, it's like a human suggestion. Isn't that interesting? It's like, human Copilot is flipping it.

#### 中文翻译:

这至少在目前是一个潜在的新工作：帮 AI 脱困。虽然随着时间推移 AI 会越来越强，可能不再需要这些人。但我喜欢这种“AI 第一，人类第二”的转变。以前是人构建，AI 辅助（比如 Copilot 刚出时只是建议一个函数）；现在反过来了，AI 构建全部，遇到不会的再找人类帮忙。这就像是“人类成了 AI 的 Copilot”，非常有意思。

---

## (01:07:15) Eric Simons

#### English:

Totally. Yeah. That's what's wild is, I think Sonnet was really the first model that flipped the equation, because that was really us, and old Cursor, and all these other things. The rapid growth started the second Sonnet went online. We actually tried building Bolt almost exactly a year ago, with the frontier models at the time. Spent a week or two building it. It just didn't work. The output, the code output was not reliable enough. It would constantly, it would be a broken app, or it would look ugly, or whatever. And then we got a sneak peek of the Sonnet stuff in May and we were like, "Oh. Okay, we should take that

project back off the shelf and green line it, because this might be it." (01:07:36): And lo and behold, that's exactly what has happened. But yeah, that's the big deal that is, kind of under the hood, this is... What's going on here is, a very critical threshold has been passed with LLM's ability to write production grade code and apps that actually look beautiful, and actually function well. It's not perfect, but there's kind of this zero to one moment that's happened where it's like, "Okay, so now, yeah. Now the AI is the first thing," and then you're kind of popping a developer in every now and then, versus the other way around.

#### 中文翻译:

完全正确。疯狂的是，我认为 Sonnet (Claude 3.5 Sonnet) 是第一个真正改变这个方程式的模型。包括我们、旧版的 Cursor 以及其他工具，爆发式增长都是从 Sonnet 上线那一刻开始的。其实大约一年前我们就尝试过构建 Bolt，用当时的尖端模型。花了一两周，但根本行不通。代码输出不够可靠，应用经常是坏的，或者很丑。直到 5 月份我们提前看到了 Sonnet 的表现，我们想：“噢，好吧，我们应该把那个项目重新拿出来启动，因为这可能就是我们要的机会。”结果正如你所见。底层的逻辑是：LLM 编写生产级代码、构建精美且功能齐全的应用的能力已经跨越了一个关键门槛。虽然不完美，但“从 0 到 1”的时刻已经发生了。现在 AI 是主力，开发者只是偶尔介入。

---

### (01:08:11) Lenny Rachitsky

#### English:

I did not know that. I didn't realize that so much of this was unlocked with, like it's sitting on top of Anthropic work, and specifically Sonnet. That was the first model, you're saying, that could code well enough.

#### 中文翻译:

我以前不知道这一点。我没意识到这么多功能的解锁是建立在 Anthropic 的工作之上，特别是 Sonnet。你是说，那是第一个编程能力足够强的模型。

---

### (01:08:22) Eric Simons

#### English:

Yeah, zero question.

#### 中文翻译:

是的，毫无疑问。

---

### (01:08:24) Lenny Rachitsky

#### English:

Wow.

#### 中文翻译:

哇。

---

### (01:08:25) Eric Simons

#### English:

Zero question. Absolutely. (01:08:25): Lenny Rachitsky: That is fascinating. Just the amount of, I don't know, revenue and business and ecommerce that that one model has unlocked, is insane. I did not realize that. (01:08:39): Eric Simons: It is. And in retrospect, I'd mentioned, we'd never done an AI product at StackBlitz, and it's tempting. Like when of ChatGPT went online, everyone started adding AI to their products. We just didn't see a clear place for a really added value. So I was not super bullish on, you know, a lot of people were like, "AGI is going to be here in 2023." You know what I mean? There's all this stuff that was being said, and I was like, "I just don't know if I necessarily buy how fast people say that it's going to move." And to a certain degree, that was the correct view. (01:09:13): What I didn't really think about though, is if AI, if LLMs are going to get better at a specific vertical, which are going to be the things that it would be. And if you look at law, for example, you want to make the best LLM for looking at case law. The problem with that stuff is, it's not deterministic. The judge's ruling is dependent on society's view of things at the time, political stuff going on, the jury. There's a ton of things, that it's not deterministic. And so you can't really create a lot of training data that's going to be super reliable, and produce really good results. And you can't just make up cases and say, "Theoretically, the judge would say this." Because, I mean, it just doesn't work. (01:09:57): Software is deterministic. When you write code and you hit run, it either runs or it doesn't. And that's the key insight Anthropic really had. They just went deep. And then, this is what they're doing, is just reinforcement learning on basically permutating every type of app you could ever build, and just spinning up tens of thousands of cores or whatever to do that. Just building tons of training data, and doing reinforcement learning, and making their LLMs the best in the world at building beautiful, reliable applications. I'm extremely bullish. It makes technical sense why, of anything, LLMs are going to get insanely better at writing code than probably most other types of applications for LLMs. Simply because it's something that can be extremely deterministic, and permuted thousands and thousands and thousands of times per second. (01:10:54): And so I think the broader trend here is... And Sonnet has woken everyone up. Google, and Open AI, all the... Everyone is now gunning for coding because, how big is the market opportunity to rewrite the software world order? It's trillions of dollars, or something, right? The world runs on software. So I think that just in a macro, the highest macro view, and why we went and raised money for Bolt. This seems like an extremely clear shot, there's this, you can kind of separate the hype of what people say and blah, blah, blah. When you break it down technically, this makes sense. It makes sense what's going on here. And for us, we're like, we are so happy to be well positioned to go and enable people to kind of ride this wave of the innovation that is here in LLMs, and is going to just keep coming, and therefore enable more people to build even crazier amazing software. So that's our world view, at least, of what's going on here at the macro.

## 中文翻译:

Lenny: 这太迷人了。仅仅一个模型就解锁了如此巨大的收入、业务和电子商务潜力，简直不可思议。Eric: 确实。回想起来，StackBlitz 以前从未做过 AI 产品。当 ChatGPT 走红时，每个人都在往产品里加 AI，但我们当时没看到明确的增值点。所以我当时并不特别看好，很多人说“2023 年 AGI 就会到来”，我不确定它是否真的会那么快。在某种程度上，那个观点是对的。但我当时没考虑到的是，如果 LLM 在某个特定垂直领域变得更强，那会是哪个领域？比如法律，你想做最好的法律 LLM，但法律是非确定性 (non-deterministic) 的。法官的裁决取决于社会观点、政治环境、陪审团等。你无法创造出大量超级可靠的训练数据。但软件是确定性的 (deterministic)。当你写完代码点击运行，它要么运行，要么不运行。这就是 Anthropic 的关键洞察。他们深入研究，通过强化学习对各种可能的应用进行排列组合，动用成千上万个核心来生成训练数据，让他们的 LLM 成为世界上构建精美、可靠应用的最强工具。我非常看好这一点。从技术逻辑上讲，LLM 在写代码方面的进步速度会远超其他应用，因为它极其确定，且可以每秒进行成千上万次的排列组合测试。所以大趋势是……Sonnet 唤醒了所有人。Google、OpenAI，所有人现在都在猛攻编程领域。因为改写软件世界秩序的市场机会有多大？那是数万亿美元的规模。世界运行在软件之上。所以从宏观来看，这也是我们为 Bolt 融资的原因。这是一个极其清晰的机会，你可以抛开那些炒作，从技术层面分析，这完全合乎逻辑。我们很高兴能处于这个位置，赋能人们利用 LLM 的创新浪潮，构建更惊人的软件。

## (01:12:05) Lenny Rachitsky

**English:**

And when did Sonnet even come out? It's been a while, right?

**中文翻译:**

Sonnet 是什么时候出来的？有一段时间了吧？

---

## (01:12:08) Eric Simons

**English:**

I think they officially, I think it was in June, when they officially put it online.

**中文翻译:**

我想是 6 月份正式上线的。

---

## (01:12:12) Lenny Rachitsky

**English:**

So since June, this is the worst it will ever be, the state of AI coding. And it's already this good. And there hasn't been anything, like they haven't launched their new model, since last June. So this tells us just how quickly things are going to start moving once they launch their next model. And as you said, everyone's gunning now for this, because they realized "We're behind on the coding piece." So, wow. This is going to get crazy.

**中文翻译:**

所以从 6 月到现在，这已经是 AI 编程“最差”的时候了，而它已经这么强了。自去年 6 月以来，他们还没发布新模型。这告诉我们，一旦下一个模型发布，事情会发展得多么快。正如你所说，现在每个人都在发力，因为他们意识到自己在编程方面落后了。哇，这会变得非常疯狂。

---

## (01:12:40) Eric Simons

**English:**

I agree. Yeah, it's been, again, there was no blog post that laid all of this out for us. It's just been kind of this- (01:12:48): Lenny Rachitsky: You just noticed the code was really good, basically. (01:12:50): Eric Simons: And from there, it's just, we've been piecing together all this other stuff. So it's been kind of the thrill of a murder mystery of, "What is going on here?" (01:12:59): Lenny Rachitsky: Oh, really. (01:13:00): Eric Simons: Yeah. You know what I mean? (01:13:01): Lenny Rachitsky: Just piecing together what you've seen on Anthropic releasing, is that what you mean? What have you been noticing, murder mystery-wise? (01:13:06): Eric Simons: Well, and then the impacts of Bolt, where we have people that are not technical using this. How are they using it? Why are they doing this? And then, all the stuff we've talked about in this podcast has been the result of nine months of just R&D and seeing the results of it, and then going, "What?" And then digging in, doing another thing and then going, "What," again. And then it just keeps happening, because there's no charted course for this.

**中文翻译:**

我同意。并没有哪篇博文为我们揭示这一切。Lenny：你只是发现代码质量变得非常好。Eric：是的，从那开始，我们一直在拼凑其他信息。这就像解开一个谋杀谜案一样刺激：“这里到底发生了什么？”Lenny：真的吗？Eric：是的。Lenny：你是说拼凑 Anthropic 发布的信息吗？你发现了什么“谜案”线索？Eric：还有 Bolt 带来的影响。我们看到非技术人员在使用它，他们是怎么用的？为什么要这么用？我们在播客里聊的所有内容，都是过去九个月研发和观察结果的产物。我们不断地感到惊讶，然后深入研究，再感到惊讶。这种循环一直在持续，因为这条路没有前人走过。

---

### (01:13:31) Lenny Rachitsky

**English:**

It's like an anthropology, if that's the right term, of just watching. It's like an emergent discovery, it sounds like. Versus you had the strategy, "Here, we're going to do this," where a persona to launch this thing will happen?

**中文翻译:**

这就像是一种“人类学观察”，如果这个词准确的话。听起来像是一种“涌现式发现”，而不是你预先设定了战略：“我们要这么做，针对这类人群发布这个产品。”

---

### (01:13:42) Eric Simons

**English:**

Yeah, a hundred percent. Yeah. I think that that's the best way to put it. It's the best way to put it. It's exactly that. And so it's very interesting to just- (01:13:54): Lenny Rachitsky: To watch, and be a part of it, I imagine. (01:13:56): Eric Simons: Yeah, yeah.

**中文翻译:**

是的，百分之百。我认为这是最好的表达方式，事实正是如此。Lenny：观察并参与其中，我想一定很有趣。Eric：是的。

---

### (01:13:57) Lenny Rachitsky

**English:**

And I think as people say, "Okay, these are just toys, they're prototypes, it's not going to work with your existing code, it's not going to scale." It's important to note just what we talked about. This is a model from last June that this is possible on, and everybody's working on the next cutting edge model that will make this even better, and that's going to come real soon. Okay. Amazing. (01:14:20): I just have a few more questions to close out this conversation. One is just, what is coming next for Bolt? What are some of the cool new features that'll be launching before this comes out? Maybe right after this comes out, maybe in mid-March?

**中文翻译:**

当人们说“这些只是玩具、只是原型，没法处理现有代码，没法规模化”时，记住我们刚才聊的很重要：这只是基于去年6月的模型。每个人都在开发下一个更强的模型，它很快就会到来。太棒了。在结束对话前我还有几个问题。一个是，Bolt接下来有什么计划？在这一集播出前后，会有哪些酷炫的新功能上线？

---

### (01:14:33) Eric Simons

## English:

Yeah. Okay. So by the time, and I'm going to go back and tell our engineers, "I said this on this podcast-"  
(01:14:33): Lenny Rachitsky: "I've committed. Sorry, guys." (01:14:40): Eric Simons: This is, I found actually being a leaky faucet on talking on podcasts and stuff, my engineer is like, "How could you tell them..." "You just have to ship it faster, now. You got to make it real," right? (01:14:49): Lenny Rachitsky: Or we'll get AI on it. (01:14:52): Eric Simons: No, no. Yeah. But I think for us it's like, again, we've seen a lot of PMs, designers, entrepreneurs, et cetera, using Bolt. And so, we're really looking at better fitting with the tools that folks are using to do those things today. Bolt's not going to replace them, or something. And if you're working at a company, how do you integrate this stuff with your existing business, or your existing product and code base? Because that's the question we often get from people is, "How do I open my," like we're talking to one of the fan companies the other day, and like, "How do we open our production code base in this, that's like 20 years old?" I'm like, "You don't. None of this stuff. That's not what you do. This is for rapid product development in your use case." (01:15:41): So the features that we're going to be shipping, I'm pretty stoked about this one, so we've been working on this for a while and we've partnered up with a company called Anima to do this. But basically, so on any Figma URL, when you're looking at a design that you've made, if you just put bolt.new in front of that URL and hit enter, it's going to suck that design into Bolt, and turn it into a full stack app or mobile app, just out of the box. (01:16:08): Lenny Rachitsky: That is genius. (01:16:09): Eric Simons: Yeah. It's- (01:16:09): Lenny Rachitsky: Amazing. (01:16:10): Eric Simons: It's going to be nuts. Yeah, I mean, it's really fun to use. Because whether you're a developer or designer, or whatever, going and taking that and turning it into an actual coded app. And the thing is, once it's in Bolt, you can just keep prompting from there. You're like, "Yeah, well, add another page here." So you can have things that, where you want pixel perfect design, you can have it, and it'll translate one to one. And it's splitting out the assets. Anima's been doing this since 2017, Figma to code. They've got the best agent in the world for, they're the number one Figma plug-in, or whatever. And so in Bolt, it's going to just work. It's just deeply integrated. (01:16:53): Lenny Rachitsky: So it's bolt.new slash, the Figma URL, to the design. (01:16:56): Eric Simons: Yep. That's all you got to do. (01:16:56): Lenny Rachitsky: Amazing. (01:16:57): Eric Simons: And in Bolt, we're going to also have a little Figma icon in the chat thing. So if you go to Bolt itself, you can click it and then paste the URL, or whatever. But yeah, but it's like that. It's from Figma to full stack app in a click. Literally. That's crazy. So that one's pretty cool. (01:17:15): And then the other one that we're working on is an integration with Slack, because often when you're talking about Figmas, like Figma links or whatever, at a company, you're in Slack or whatever. So you're having conversations about, "Hey, we should really add a page to this that does da-da-da." And so we're actually creating a Bolt Slack bot whose job is to basically act like a developer on your team. And so you can be, in a thread, you can be like, "Hey, I think we should a homepage." "Yeah, okay. @Bolt, can you whip this up real quick?" And it'll go and just suck down the entire conversation history so far in your Slack, on the thread or whatever. (01:17:54): You'll be like, "Okay, cool. So you kind of want me to take this Figma URL, which I can convert automatically," thanks to the future I just mentioned right before this. "I can just go and convert that thing out of the box, and then you want me to add a page to it, and then do this thing. Got it. I'm going to go do that. Oh, here's the URL where you can open it up and keep prompting." You know what I mean? So it's like having a developer or somebody to kind of kick this thing out. And you're just like, "Go make this thing real quick." (01:18:15): So those two things, I think it's, again. You start think about how our company's going to change how they're currently doing product development. And even just like, "Hey, we need to spin up a marketing site. Here's the thing." We're like, "Can you do that, Bolt?" "Yep. Let me go do that." You know what I mean? And that's why I'm kind of excited about those two, in particular, because I think it's going to be well received, and folks are going to be stoked about it. I hope. Knock on wood.

## 中文翻译:

好的。我要回去告诉工程师们：“我在播客里把事儿说了。” Lenny：“我已经承诺了，抱歉伙计们。” Eric：我发现自己在播客里总是管不住嘴，工程师会说：“你怎么能告诉他们……”“那你们只能赶快把它做出来了。” Lenny：或者让 AI 来做。Eric：不不。我们看到很多 PM、设计师和创业者在用 Bolt，所以我们正致力于更好地适配大家现有的工具。Bolt 不是要取代它们。如果你在一家公司工作，如何将它与现有业务或 20 年前的老代码库集成？人们常问：“我怎么在这里打开 20 年前的生产代码库？”我会说：“你别这么干，这不是它的用途。它是为了快速产品开发。”我们即将推出的功能里，我最兴奋的是这个：我们与一家叫 Anima 的公司合作。基本上，在任何 Figma 设计稿的 URL 前面加上 `bolt.new/` 然后回车，它就会把设计稿吸进 Bolt，直接生成一个全栈应用或移动应用。Lenny：这太天才了。Eric：是的，这会很疯狂。它非常好用，无论你是开发者还是设计师，都能把设计变成真实的代码应用。一旦进入 Bolt，你就可以继续通过提示词修改：“在这里加个页面。”你可以拥有像素级的精确设计，它会 1:1 转化。Anima 自 2017 年起就在做 Figma 转代码，他们有世界上最好的智能体。在 Bolt 里，这一切将深度集成。Lenny：所以就是 `bolt.new/` 加上 Figma 的 URL。Eric：没错。Bolt 里也会有一个 Figma 图标，点击粘贴即可。从 Figma 到全栈应用，只需一键。另一个功能是 Slack 集成。在公司里，大家经常在 Slack 讨论设计稿：“嘿，我们应该加个页面做某某事。”我们正在创建一个 Bolt Slack 机器人，它的角色就像你团队里的开发者。在讨论串里，你可以说：“@Bolt，能把这个做出来吗？”它会读取 Slack 里的对话历史，理解你的意图（包括自动转换提到的 Figma 链接），然后说：“好的，我这就去把设计稿转成应用并加上那个页面。这是链接，你可以打开继续修改。”这就像有个开发者随时待命帮你把想法变成现实。这两点将改变公司目前的产品开发方式。

---

## (01:18:40) Lenny Rachitsky

### English:

Those are awesome features. I love the Slack piece, because when I think about agents, there's always talk about agents. To me the simplest way to understand it is just a Slack bot, just that AI can talk to you like they're a person in Slack. And I love that's exactly what you're doing. And this is this gigantic feature of just like, "Hey, you just have this engineer now that can go build stuff for you." (01:18:59): Let me actually ask you a question along these lines, that I was meaning to ask, but I forgot. Is just, your engineering team, what are they using to build Bolt? I imagine it's a lot of Cursor. How much is Bolt, at this point, involved in building Bolt? And is there any other tools that they found useful, find useful, that are worth highlighting?

### 中文翻译:

这些功能太棒了。我特别喜欢 Slack 那个，因为谈到“智能体”(agents)，最简单的理解方式就是 Slack 机器人，AI 像真人一样在 Slack 里和你交流。这就像你现在有了一个随时帮你构建东西的工程师。顺便问个我刚才忘了问的问题：你们的工程团队用什么来构建 Bolt？我猜有很多 Cursor。目前 Bolt 在构建 Bolt 自身的过程中参与度有多高？还有其他值得推荐的工具吗？

---

## (01:19:18) Eric Simons

### English:

Yeah, good question. Yeah, we definitely use Cursor. Our folks use Cursor a lot. We use Bolt a lot for the product development process, like a ton, we're using it. And we're doing basically the flow that I described, where if things that need to be Pixel perfect, we're going to Figma for. And often we're taking that and we're pulling that into Bolt, because we've got access to the integration today. So pull that into Bolt and we're saying, "Hey, go add these things or whatever." Or just saying, "Hey, here's a screenshot of our UI, go do da-da-da." (01:19:47): Other AI tools that the developers are using. I think those are the primary ones. I mean, I think we've got a subscription to Claude, and ChatGPT, and things like that. But I think for development, Cursor is the main thing,

## 中文翻译:

好问题。我们确实大量使用 Cursor。在产品开发过程中，我们也大量使用 Bolt。基本上就是我描述的流程：需要像素级精确的东西先在 Figma 做，然后拉进 Bolt（我们内部已经可以用那个集成功能了），接着告诉它“去添加这些功能”。或者直接给它一张 UI 截图说“照着这个做某某修改”。其他 AI 工具的话，主要是订阅了 Claude 和 ChatGPT。但对于开发来说，Cursor 是主力。

---

## (01:20:03) Lenny Rachitsky

### English:

Yeah, it's cool how few tools, like there's so many AI tools, and it's interesting how few people actually end up using. It's like Cursor, Claude, ChatGPT, and then maybe another tool. Like Bolt.

### 中文翻译:

是的，虽然 AI 工具多如牛毛，但大家最后真正常用的就那么几个：Cursor、Claude、ChatGPT，再加上像 Bolt 这样的工具。

---

## (01:20:13) Eric Simons

### English:

Yeah. Totally.

### 中文翻译:

没错。

---

## (01:20:14) Lenny Rachitsky

### English:

Final-ish question. Say somebody is opening up Bolt for the first time. What's something that, imagine you could sit next to every new user that's just trying Bolt for the first time, and you could whisper a tip in their ear to be successful with Bolt. What would that tip be?

### 中文翻译:

最后一个问题。假设有人第一次打开 Bolt。想象一下如果你能坐在每个新用户身边，在他们耳边轻声说一个能让他们成功使用 Bolt 的秘诀，那会是什么？

---

## (01:20:33) Eric Simons

### English:

And this is like, because we have a lot of different types of users. I imagine you're talking about PMs or designers, and that sort of- (01:20:39): Lenny Rachitsky: Let's do PMs. That's a good one. That's a lot of the audience here. (01:20:41): Eric Simons: I would say, talk to this thing like you do a Linear ticket, or a JIRA ticket. That would be my advice. And talk to this like you would, like you're talking to one of the developers on your team. And what that means is, be specific on things that matter. And on things where, also, you can let it be creative. You can go to and just say, "Hey, make it prettier." And it does a good job, it actually does a really good job, when you give it just, vibes. So anyway, I think for PMs it's like, you have the skillset. You know how to do this. This is, just think of this as your coworker, your developer coworker.

## 中文翻译:

这取决于用户类型。我猜你指的是 PM 或设计师……Lenny：就说 PM 吧，这是我们听众的主力。Eric：我会说，像写 Linear 或 JIRA 工单那样跟它交流。像跟你团队里的开发者说话一样。这意味着：在关键的事情上要具体、明确；而在某些地方，你也可以让它发挥创意，比如直接说“让它变漂亮点”。当你只给它一种“感觉”(vibes) 时，它其实做得非常好。所以对于 PM 来说，你已经具备了所需的技能，你懂得如何沟通。就把 Bolt 当作你的开发者同事。

---

## (01:21:24) Lenny Rachitsky

### English:

I love that. Because these tools are so easy, you just go in and it's like tell it a thing, and then, cool. You have a website. It's coded, boom, done. And what I'm hearing here is, take a little time to craft your ask. It may be tempting to just start, "Cool. Build me a serum," and then you're stuck with that first version. And then you're like, "Oh, well, okay. I didn't mean that." So it sounds like your advice is take some time to craft the ask, and be clear about what you want.

### 中文翻译:

我喜欢这个建议。因为这些工具太简单了，你进去随口说一句，砰，网站就出来了。但我听到的是：花点时间打磨你的请求。虽然随口说“给我做个 CRM”很有诱惑力，但你可能会被第一个版本卡住，然后发现“噢，这不是我想要的”。所以建议是花时间构思请求，明确你的目标。

---

## (01:21:49) Eric Simons

### English:

Yeah, totally. Especially if you have a clear vision of what you're trying to build. And something reasonably sophisticated. And what I recommend everyone to do, if it's your first time trying Bolt and you're like, "What should I have it do, and I don't have an idea"? Tell it to build you a personal website. There's something like magic. You take your LinkedIn copy, and paste your LinkedIn bio and work experience, just like select text, copy, paste it. "I need a website, my name is so-and-so, here's my LinkedIn history. My favorite color is blue, and I like dogs." And then hit paste, right? (01:21:49): Lenny Rachitsky: "Make it prettier." (01:22:24): Eric Simons: [inaudible 01:22:24] actor. Yeah. And then, you know what I mean? And then you can hit deploy. And if you don't have a .com yet, now you can. Right? I mean, now you have a real, personalized service. I think there's kind of a moment around that where it's like, "Oh, okay, wow. This," it's zero shot, zero shot 99.999% of the time. You're getting a beautiful personal website that you didn't have before, that would've taken you an hour, on Wix. If not more. And that gives you the taste of, "Oh, okay, cool. So if I really take the time to think this through, and make a PRD, and then put that in piece by piece into this thing, the sky's the limit."

### 中文翻译:

是的，完全正确。特别是如果你有清晰的愿景或要做复杂的东西。我建议第一次用 Bolt 但没主意的人：让它帮你做一个个人网站。这很有魔力。你把 LinkedIn 上的个人简介和工作经历复制下来，直接粘贴进去说：“我需要一个网站，我叫某某，这是我的经历。我喜欢蓝色，喜欢狗。”然后按回车。Lenny：“再漂亮点。”Eric：没错。然后点击部署。如果你还没有个人域名，现在就可以拥有一个真实的、个性化的服务。那一刻你会意识到：“哇，这几乎是‘一语即成’。”你得到了一个以前没有的精美网站，而这在 Wix 上可能要花一小时甚至更久。这会让你尝到甜头：“如果我真的花时间思考，写个 PRD，然后一步步喂给它，那简直无可限量。”

---

**English:**

Eric, this has been just insane on so many levels. I have so much to process, I think a lot of listeners do too. Maybe as an actual final question, I saw this story about how when you were starting StackBlitz, or maybe even before StackBlitz, you squatted in the AOL office. Because you had some badge that still worked. Maybe just tell that story.

**中文翻译:**

Eric, 这一集在很多层面上都太疯狂了。我有太多东西需要消化，我想听众也是。作为真正的最后一个问題，我听过一个故事：在你创立 StackBlitz 甚至更早之前，你曾“蜗居”在 AOL 的办公室里，因为你有个还能用的工牌。能讲讲那个故事吗？

---

**(01:23:18) Eric Simons**

**English:**

Yeah, that was the thing I was most known for. That happened like 2012, and I was 19 years old, so it's been a very long time. I think I'm 33, now or something. (01:23:27): Lenny Rachitsky: The statutes of limitations are expended. (01:23:30): Eric Simons: Yeah, yeah, exactly. But yeah, for a while I was like, "I have to do something more notable than living at AOL. This can't be what I'm known for." But now people are like, "Oh, wait. You're the StackBlitz guy, and you're the AOL guy?" (01:23:43): So when I first came out to Silicon Valley, we got into a, I was building a K12 education startup at the time, and this is back during the days where Y Combinator, they only gave you like 20 grand, and so there was this offshoot of Y Combinator called Imagine K12, it was Jeff Ralston who, I think he was CEO of YC a couple of years back. And then Imagine K12 merged into Y Combinator a couple of years after Rob went through it, but anyways, they had an office space at AOL. At the time, AOL was trying to reinvigorate the company and they were like, "Hey, we should get young startup blood in here, so let's rent out office space to teenagers," basically. (01:24:24): So I was there, and we ended up running out of money. 20 grand doesn't go very far in the Valley, so three or four months in, we were like, "Oh God, what do we do?" And I was going to the AOL office multiple times a week, because we had access cards to get in to get to the investors' offices. And I realized, I was like, "You know, they have couches here, and they have food. There's ramen that you can microwave, and there's a gym where there's a shower, and even you can do laundry." And then, so I was like, "I don't know, maybe while I figure this out, I'll just live out of here." And so that's what I ended up doing for, I think, four or five months. I was living out of this headquarters over on Page Mill in El Camino, in Palo Alto. (01:25:12): And then, I got away with it for a while just because the guards, the security guards, they worked 12-hour shifts. And so the guys that, when I was there at night... And I was coding, all day every day, basically. So the guys at night just were like, "Dang, this guy works really hard." And then in the morning they'd be like, "Wow, this guy is working really hard." And I became friends with some of them, and then eventually, I think there were also a whole bunch of Stanford students that I think they put bunks in one of the aisles. It was just started getting out of control, so I think they started cracking down. And then one morning, at like 4:00 in the morning, a guard came in and threw me out. (01:25:52): I'm from Chicago. I don't know anyone. At that point I'm like, "I know no one in the Bay Area." So I went to a Starbucks, which was not open. I slept on the table outside of the thing. And I think I hit up one of the other entrepreneurs that was in the program. I was like, "Do you have a couch? I think I kind of need it, at this point." Yeah, the press got wind of it, and it was this worldwide story. But I lived on a dollar a day. That was the crazy thing. My burn rate was a dollar a day, at that time.

**中文翻译:**

是的，那是我以前最出名的事。那是 2012 年，我才 19 岁，已经过去很久了，我现在都 33 岁了。Lenny：追诉期已经过了。Eric：没错。有一段时间我想：“我得做点比‘住在 AOL’更出名的事，不能一辈子顶着这个标签。”但现在人们会说：“噢等等，你就是那个 StackBlitz 的家伙，也是那个 AOL 的家伙？”当时我刚来硅谷，在做一个 K12 教育创业项目。那时候 YC 只给 20000 美元。当时 YC 有个分支叫 Imagine K12，他们在 AOL 的办公楼里有空间。当时 AOL 想焕发活力，决定把办公室租给年轻人。后来我们的钱花光了，2 万美元在硅谷撑不了多久。三四个月后我们陷入困境。我当时每周去 AOL 办公室好几次，因为我有门禁卡。我发现那里有沙发、有食物（可以微波炉加热的拉面）、有健身房可以洗澡，甚至能洗衣服。我想：“也许在想到办法前，我可以住在这儿。”于是我真的在那儿住了四五个月，就在帕罗奥图（Palo Alto）的总部。我之所以没被发现，是因为保安是 12 小时轮班。晚班保安看到我在写代码，觉得“这家伙真勤奋”；早班保安看到我还在写代码，也觉得“这家伙真勤奋”。我还和一些保安成了朋友。后来有一群斯坦福学生也跑来在走廊里搭床位，事情开始失控，公司开始清理。一天凌晨 4 点，保安把我赶了出去。我来自芝加哥，在湾区举目无亲。我去了家还没开门的星巴克，睡在室外的桌子上。后来我联系了项目里的另一个创业者问能不能借宿沙发。后来媒体报道了这件事，成了全球新闻。最疯狂的是，我当时每天的生活费只有 1 美元。

---

## (01:26:22) Lenny Rachitsky

**English:**

What did you use that dollar for?

**中文翻译:**

那 1 美元你用来买什么？

---

## (01:26:25) Eric Simons

**English:**

This is back when McDonald's had the Dollar Menu. Literally. So it was like, I occasionally would go and get a cheeseburger or whatever. Yeah, it was ultimate scrappiness.

**中文翻译:**

那时候麦当劳还有“1 美元菜单”。真的，我偶尔会去买个芝士汉堡之类的。那是极度的艰苦奋斗。

---

## (01:26:41) Lenny Rachitsky

**English:**

You're technically homeless. From homeless, to one of the fastest growing startups in history. Eric, what a journey. This is such an interesting point in time of your life, and of just tech. No matter what happens, I'm sure you'll be extremely successful, but it's such an interesting just point in that journey. And I'm thankful that you made time to share it with us.

**中文翻译:**

你当时技术上属于无家可归。从无家可归到拥有历史上增长最快的初创公司之一。Eric，多么精彩的旅程。这是你人生中、也是科技史上一个非常有趣的时刻。无论未来发生什么，我相信你都会非常成功。感谢你抽出时间与我们分享。

---

## (01:27:01) Eric Simons

**English:**

It's always good to just have the perspective of, you should start companies to keep the mindset that you're doing it to have fun. So, stoked to see where this goes, one way or the other. It's going to be interesting.

**中文翻译:**

保持一种“创业是为了好玩”的心态总是好的。无论结果如何，我都非常期待未来的走向，这一定会很有趣。

---

### **(01:27:16) Lenny Rachitsky**

**English:**

Eric, final questions. Where can folks find online if they want to reach out, maybe follow up on some stuff you shared, and how can listeners be useful to you?

**中文翻译:**

Eric，最后的问题。如果大家想联系你或跟进你分享的内容，可以在哪里找到你？听众们能为你做些什么？

---

### **(01:27:23) Eric Simons**

**English:**

Yeah, totally. I mean, yeah, so bolt.new is the website, over on Twitter, I'm @ericsimons40 on Twitter, and I think our Twitter account is @boltdotnew, not with a period. It's like, B-O-L-T-D-O-T-N-E-W. And yeah, I'm curious to hear what folks think. I mean, this is, again, we are learning so much from the people that are coming and trying this thing out and giving their feedback. And within the first meeting of it going online, we were not the experts on how use the tool anymore, and it's been that way ever since. And so, I love hearing from folks on what they want to see next, and how this is helping them. And where they run into problems, like where we need to go and fix things. So my email address is Eric@stackblitz.com. That's Eric with a C. So I'd love to hear from anyone, whether it's a DM on Twitter, or an email.

**中文翻译:**

好的。网站是 bolt.new，推特账号是 @ericsimons40，Bolt 的官方推特是 @boltdotnew (拼写出来)。我很想听听大家的想法。我们从尝试产品并提供反馈的用户身上学到了太多。产品上线后的第一分钟，我们就已经不再是“最懂这个工具”的专家了，用户才是。所以我很想听听大家希望看到什么新功能，以及它如何帮助了你们，或者遇到了什么问题。我的邮箱是 Eric@stackblitz.com (Eric 是 C 结尾)。无论是推特私信还是邮件，我都非常欢迎。

---

### **(01:28:18) Lenny Rachitsky**

**English:**

Amazing. Eric, thank you so much for being here.

**中文翻译:**

太棒了。Eric，非常感谢你能来。

---

### **(01:28:21) Eric Simons**

**English:**

Awesome. Thank you so much for having me. This is a blast.

**中文翻译:**

太好了。非常感谢邀请我，聊得很开心。

---

### **(01:28:23) Lenny Rachitsky**

**English:**

Bye, everyone. (01:28:26): Thank you so much for listening. If you found this valuable, you can subscribe to the show on Apple Podcasts, Spotify, or your favorite podcast app. Also, please consider giving us a rating or leaving a review, as that really helps other listeners find the podcast. You can find all past episodes or learn more about the show at [lennyspodcast.com](http://lennyspodcast.com). See you in the next episode.

**中文翻译:**

大家再见。感谢收听。如果你觉得本期节目有价值，可以在 Apple Podcasts、Spotify 或你喜欢的播客应用中订阅。也请考虑给我们评分或留下评论，这能帮助更多听众发现我们。你可以在 [lennyspodcast.com](http://lennyspodcast.com) 找到往期所有节目或了解更多信息。下期见。