

RYAN SINGER

LENNY'S PODCAST

BILINGUAL TRANSCRIPT

ORIGINAL BY

Lenny Rachitsky

@lennysan • x.com/lennysan

ANALYSIS BY

@Penny777 • x.com/penny777

Ryan Singer - 双语对照

Lenny's Podcast: Ryan Singer - Shape Up Method

00:00:00 Ryan Singer

English:

I often use this analogy of if you're doing a home renovation, you can have the most beautiful rendering of the new bedroom and we're going to have these lamps on the side of the bed that are coming out from the wall. But if you haven't checked if there's electricity in that wall there or not, it's going to drastically change the cost and the time and everything. What we need to do in a shaping session is we come out with some kind of diagram where engineers, product and design, they're saying, "We understand that." So the first thing is we are not going to start something unless we can see the end from the beginning. We're not going to take a big concept and then say, "What's the estimate for this thing?" We're going to go the other way around and we're going to say, what is the maximum amount of time we're willing to go before we actually finish something? How do we come up with a idea that's going to work in the amount of time that the business is interested in spending?

中文翻译:

我经常用装修房子来做类比：你可以为新卧室画出最精美的效果图（Rendering），设计那种从墙上伸出来的床头灯。但如果你没检查那面墙里到底有没有电线，这将会剧烈地改变成本、时间和所有计划。在“定型”（Shaping）环节中，我们需要产出一种图表，让工程师、产品和设计人员都能达成共识说：“我们理解了。”所以首要原则是：除非我们能“一眼望到头”（看到从开始到结束的全过程），否则我们不会启动。我们不会拿一个宏大的概念然后问：“这东西估时多久？”我们会反其道而行之，我们会问：在真正完成某件事之前，我们愿意投入的最长时间是多少？我们如何想出一个点子，能在公司愿意花费的时间范围内顺利完工？

00:00:54 Lenny Rachitsky

English:

Today my guest is Ryan Singer. Ryan was one of the first few hires at 37signals, and through his experience of building Basecamp and 17 years of building product at 37signals, he wrote a book called Shape Up, which shares a very different approach to building software. Appetites instead of deadlines. A big focus on bringing design engine product together into a room to shape the plan versus writing long PRDs or trying to finalize designs before you start building. I've noticed more and more teams adopting the Shape Up method, and especially with AI starting to change how we work and build product, there's this shift coming in how product teams will operate. And so I thought this was the perfect time to do a deep dive into the Shape Up method.

中文翻译:

今天的嘉宾是 Ryan Singer。Ryan 是 37signals 的早期员工之一。凭借开发 Basecamp 的经验以及在 37signals 17 年的产品开发历程，他写了一本名为《Shape Up》的书，分享了一种截然不同的软件开发方法。它提倡用“投入意愿”（Appetites）代替“截止日期”（Deadlines）。它非常强调将设计、工程和产品人员聚在一起共同“定型”方案，而不是撰写冗长的 PRD（产品需求文档）或在开工前试图敲定所有设计细节。我注意到越来越多的团队开始采用 Shape Up 方法，特别是在 AI 开始改变我们工作和构建产品方式的当下，产品团队的运作模式正面临转型。因此，我认为现在是深度探讨 Shape Up 方法的最佳时机。

00:02:14 Lenny Rachitsky (Sponsor: WorkOS)

English:

This episode is brought to you by WorkOS. If you're building a SaaS app, at some point your customers will start asking for enterprise features like SAML authentication and SCIM provisioning. That's where WorkOS comes in, making it fast and painless to add enterprise features to your app. Their APIs are easy to understand so that you can ship quickly and get back to building other features. Today, hundreds of companies are already powered by WorkOS, including ones you probably know like Vercel, Webflow and Loom.

中文翻译:

本期节目由 WorkOS 赞助。如果你正在构建 SaaS 应用，你的客户迟早会要求提供企业级功能，比如 SAML 身份验证和 SCIM 供应（SCIM provisioning，一种用于自动化用户账号管理的标准）。这就是 WorkOS 的用武之地，它能让你快速且无痛地为应用添加企业级功能。他们的 API 易于理解，让你能快速交付并专注于核心功能的开发。如今，已有数百家公司由 WorkOS 提供支持，包括你可能熟悉的 Vercel、Webflow 和 Loom。

00:03:33 Lenny Rachitsky (Sponsor: Merge)

English:

This episode is brought to you by Merge. Product leaders, yes, like you, cringe when they hear the word integration. They're not fun for you to scope, build, launch or maintain, and integrations probably aren't what led you to product work in the first place? Lucky for you the folks at Merge are obsessed with integrations. Their single API helps SaaS companies launch over 200 product integrations in weeks, not quarters.

中文翻译:

本期节目由 Merge 赞助。产品负责人——没错，说的就是你——听到“集成”（Integration）这个词时都会感到头大。对你来说，确定集成的范围、构建、发布或维护它们毫无乐趣，而且集成可能根本不是你投身产品工作的初衷。幸运的是，Merge 的团队对集成非常痴迷。通过他们的统一 API，SaaS 公司可以在几周内（而不是几个季度）发布超过 200 项产品集成。

00:04:42 Lenny Rachitsky

English:

Ryan, thank you so much for being here. Welcome to the podcast. I think this is going to be a legendary episode. There's a lot of interest these days in different ways of working, especially ways that are Agile and SAFe and Scrum and all these ways that people hear about working. Especially in this world of AI where everything's just changing. It feels like there's just an increased interest in exploring different ways of working and specifically it feels like there's been a rise in interest in Shape Up the stuff that you talk about.

中文翻译:

Ryan, 非常感谢你能来。欢迎来到播客。我觉得这会是传奇的一集。如今人们对不同的工作方式非常感兴趣, 尤其是那些大家耳熟能详的敏捷 (Agile)、SAFe (规模化敏捷框架) 和 Scrum 等等。特别是在 AI 改变一切的当下, 大家似乎更有兴趣去探索不同的工作模式, 尤其是你所倡导的 Shape Up 方法, 关注度明显在上升。

00:05:38 Ryan Singer

English:

Yeah, I think it's interesting that we're talking now. I mean, the book came out in 2019 and it's, I've been hearing more and more like, "Oh, we know somebody who's trying it or we're hearing it when we go talk to other companies." So, I think, it's a wave that's slowly building. And it's funny, when it came out, I even tried to have an online forum to get everyone who's interested to talk together and what I started to learn pretty early on is that people don't like to talk about their struggles shipping. Especially CPOs and CTOs don't like to go on a public forum and say, "Our company isn't shipping or our engineering team is stuck, or our team is always lost in the weeds." That's not an easy community topic on an online forum.

中文翻译:

是的, 我觉得现在聊这个很有意思。这本书是 2019 年出版的, 但我现在听到越来越多的反馈, 比如“哦, 我们认识的人正在尝试它”或者“我们在与其他公司交流时听到了它”。所以, 我认为这是一个正在缓慢积聚的浪潮。有趣的是, 书刚出版时, 我甚至尝试建立一个在线论坛, 让感兴趣的人一起交流, 但我很快发现: 人们不喜欢谈论他们在交付 (Shipping) 方面的挣扎。尤其是 CPO 和 CTO, 他们不喜欢在公共论坛上说: “我们公司交不出东西”、“我们的工程团队卡住了”或者“我们的团队总是陷入细节泥潭 (Lost in the weeds)”。在线论坛上, 这并不是一个轻松的社区话题。

00:07:12 Lenny Rachitsky

English:

And there aren't many options for how to build product. All people really read about is Scrum/Agile/SAFe as they scale and then there's Waterfall, which, "No, I never do Waterfall." Then there's the start up way of just ship and maybe one or two week cycles and then there's Shape Up, so it feels like it's one of the rare other options that exists.

中文翻译:

而且关于如何构建产品, 其实并没有太多选择。大家读到的通常只有 Scrum、敏捷、或者规模化后的 SAFe, 然后就是被大家嫌弃的瀑布流 (Waterfall) —— “不, 我绝不瀑布流”。再就是创业公司那种“只管交付”的模式, 可能是一两周一个周期。接着就是 Shape Up 了, 所以它感觉是现有的极少数替代方案之一。

00:08:16 Ryan Singer

English:

Well, I mean, that came as a surprise to me. I mean, when I wrote the book, I had been in Basecamp at that time, I think 15 years, and I actually didn't even know the outside world. I mean, it was Jason's idea to even write the book. Because he said, "Look, a lot of people are going to want to know about this. A lot of people are struggling." And I'm like, "Well, okay." I knew our inside story of we had some growing pains and we had to be able to formalize the way that we were working and shipping so that as we brought new

people in that they could participate in that and we could stay fast. So, I knew our internal struggles, but I honestly didn't know anything about the outside world.

中文翻译:

这对我来说其实挺意外的。写书的时候，我在 Basecamp 已经待了大概 15 年，我其实并不了解外面的世界。写这本书甚至是 Jason（37signals 创始人）的主意。他说：“听着，很多人会想了解这个，很多人都在挣扎。”我当时想：“好吧。”我了解我们内部的情况，我们经历过成长的烦恼，必须将工作和交付的方式正式化，这样当新成员加入时，他们才能参与进来，我们也能保持高效。所以我了解内部的挣扎，但坦白说，我对外界一无所知。

00:09:24 Ryan Singer

English:

Yeah. Well, my wife heard me saying the same thing over and over again on every phone call. And she's overhearing me and she's like, "You have to make a course, you have to do something. You always are saying the same thing." So then this led to this course that we made, which is called Shaping in Real Life. And well, yeah, the idea is the real life part, right? How do I make this work if my designers don't code? It's very contentious to get engineering time. You know what I mean? When there's all these different pressures that Basecamp didn't have.

中文翻译:

是的。我妻子听我在每次电话里反复说同样的话，她听得受不了了，就说：“你得办个课程，你得做点什么，你总是在重复同样的内容。”于是就有了我们现在做的这个课程，叫《现实生活中的定型》（Shaping in Real Life）。核心点就在于“现实生活”这部分，对吧？如果我的设计师不会写代码，我该怎么推行？争取工程师的时间非常困难。你懂我的意思吗？当面临各种 Basecamp 以前从未遇到过的压力时，该怎么办。

00:10:07 Ryan Singer

English:

I think the way it's different starts with how the way we were working was a little bit different. So, I started working with Jason and David on the first version of Basecamp, which was the flagship product of 37signals back in 2003. We were a team of three. And, I mean, I think, it's for any really small team, when you're just starting out, you don't need a process. You don't need a way of working. It just happens organically because you're together. You don't have to explain it to other people, it just happens on its own, right? But there was always this really intense urgency from both Jason and David, "We've got to get to something we can ship. We have to finish this and move on. We have to get to something that's done."

中文翻译:

我认为这种差异始于我们工作方式的不同。2003 年，我开始和 Jason 以及 David（DHH）一起开发 Basecamp 的第一个版本，那是 37signals 的旗舰产品。当时我们只有三个人。对于任何极小的团队来说，刚起步时不需要流程，不需要特定的工作方式。因为大家在一起，一切都是自然而然发生的。你不需要向别人解释，它就这么运行了，对吧？但 Jason 和 David 总是有一种强烈的紧迫感：“我们必须做出能交付的东西。我们必须完成这个然后继续下一个。我们必须达到‘完工’的状态。”

00:11:32 Lenny Rachitsky

English:

Let me actually ask about this because this is really interesting. So this is DHH. He was working part-time when he started 37signals. Was he working on Ruby basically and that whole thing?

中文翻译:

我想追问一下，这非常有意思。所以那是 DHH（David Heinemeier Hansson）。他在 37signals 创立初期是兼职工作的？他当时主要是在研究 Ruby 和那一整套东西吗？

00:11:45 Ryan Singer

English:

Well, Rails came out of the first... So, he told Jason, "I want to try building this in Ruby," because before they had done some collaboration and David had done things in PHP before that and he had this new idea, he wanted to try Ruby, this language he fell in love with. And then the framework, Ruby on Rails, he ended up releasing that after Basecamp was standing. Because it was extracted from the things that were necessary to give V1 of Basecamp to stand up.

中文翻译:

嗯，Rails 是从第一个版本中诞生的……他当时告诉 Jason：“我想尝试用 Ruby 来构建这个，”因为之前他们合作过，David 用的是 PHP，但他有了新想法，想尝试 Ruby，那是他爱上的语言。而 Ruby on Rails 这个框架，是他最终在 Basecamp 运行起来之后才发布的。因为它是从构建 Basecamp V1 所需的必要组件中提取出来的。