

STEWART BUTTERFIELD

LENNY'S PODCAST

BILINGUAL TRANSCRIPT

ORIGINAL BY

Lenny Rachitsky

@lennysan • x.com/lennysan

ANALYSIS BY

@Penny777 • x.com/penny777

Stewart Butterfield - 双语对照

This is the complete bilingual transcript for Lenny's Podcast featuring Stewart Butterfield.

[00:00:00] Stewart Butterfield

English:

This is 2014. That was the year that Slack actually launched. I was interviewed by MIT Technology Review and asked if we were working to improve Slack. I said, "I feel like what we have right now is just a giant piece of shit. It's just terrible and we should be humiliated that we offer this to the public." To me that was like, "You should be embarrassed." If you can't see almost limitless opportunities to improve, then you shouldn't be designing the product.

中文翻译:

那是 2014 年，也是 Slack 正式发布的年份。当时我接受了《麻省理工科技评论》的采访，被问到我们是否在努力改进 Slack。我说：“我觉得我们现在的产品简直就是一坨巨大的狗屎。它太糟糕了，把它公之于众简直让我们感到羞愧。”对我来说，那种感觉就是：“你应该感到无地自容。”如果你看不见几乎无限的改进空间，那么你根本就不该去设计产品。

[00:00:24] Lenny Rachitsky

English:

Slack was famous for being one of the early, consumerized B2B SaaS products.

中文翻译:

Slack 以早期“消费级”B2B SaaS 产品的代表而闻名（指将 C 端产品的易用性带入企业级软件）。

[00:00:29] Stewart Butterfield

English:

At more than one company all hands, I made everyone in the company repeat this as a chant. In the long run, the measure of our success will be the amount of value that we create for customers, and you can put effort into demonstrating that you have created this value and stuff like that, but there's no substitute for actually having created it.

中文翻译:

在不止一次的公司全员大会上，我让公司里的每个人都像喊口号一样重复这句话：从长远来看，衡量我们成功的标准将是我们为客户创造的价值。你可以花精力去证明你已经创造了这些价值，但任何东西都无法替代“真正创造价值”本身。

[00:00:45] Lenny Rachitsky

English:

Something else I heard that you often espouse is friction in a product experience is actually often a good thing?

中文翻译:

我还听说你经常拥护的一个观点是：产品体验中的“摩擦力”（指阻碍或步骤）实际上往往是一件好事？

[00:00:52] Stewart Butterfield

English:

It became an assumption that it should always be trying to remove friction when the challenge is really comprehension. If your software stops me and asks me to make a decision and I don't really understand it, you make me feel stupid. If people could get over the idea of reducing friction as a number of goal or reducing the number of clicks or taps to do something, and instead focus on how can I make this simple? How do I prevent people from having to think in order to use my software?

中文翻译:

人们往往默认应该始终尝试消除摩擦，但真正的挑战其实在于“理解力”。如果你的软件停下来让我做一个决定，而我并不真正理解它，你就会让我觉得自己很蠢。如果人们能不再把“减少摩擦”或“减少点击/跳转次数”作为首要目标，而是专注于“我该如何让它变得简单？”以及“我该如何让人们在使用我的软件时不必思考？”（即 Don't Make Me Think 原则）。

[00:01:15] Lenny Rachitsky

English:

You started two companies, both famously pivoted. I imagine many people come to you for advice on pivoting.

中文翻译:

你创办了两家公司，两家都经历了著名的转型（Pivot）。我想很多人都会找你咨询关于转型的建议。

[00:01:20] Stewart Butterfield

English:

The decision is about have you exhausted the possibilities? Creating the distance so that you can make an intellectual rational decision about it rather than an emotional decision is essential. And the reason I say you have to be coldly rational about it is because it's fucking humiliating.

中文翻译:

转型的决定在于：你是否已经穷尽了所有的可能性？拉开距离，从而做出理智的决策而不是情感化的决策，这是至关重要的。我之所以说你必须保持冷酷的理智，是因为承认失败真的太他妈丢脸了。

[00:01:36] Lenny Rachitsky

English:

Today, my guest is Stewart Butterfield, a founder and product legend who rarely does podcasts. Stewart founded Flickr and then Slack, which he sold to Salesforce in one of the biggest acquisitions in tech history at the time. There is so much product and leadership wisdom locked away in his head. I feel like our conversation just scratched the surface. We chat about utility curves, something he calls the owner's delusion, a hilarious pattern he sees at companies he calls hyperrealistic work-like activities, what he's learned about product and craft and taste and Parkinson's law, why you need to obsess with not making your users think, the backstory on his legendary we don't sell saddles here memo, and so much more. A huge thank you to Noah Weiss, Chris Cordell, Ali Rael, and Johnny Rogers for suggesting topics and questions for this conversation. This is a really special one and I really hope to have Stewart back to delve even deeper.

中文翻译:

今天的嘉宾是 Stewart Butterfield，一位极少参加播客的创始人兼产品界传奇人物。Stewart 先后创办了 Flickr 和 Slack，后者以当时科技史上规模最大的收购案之一卖给了 Salesforce。他的脑子里锁着太多的产品和领导力智慧。我觉得我们的谈话仅仅触及了皮毛。我们聊到了效用曲线（Utility Curves）、他称之为“所有者错觉”（Owner's Delusion）的概念、他在公司中观察到的一种搞笑模式——“超写实类工作活动”（Hyperrealistic work-like activities）、他在产品、工艺、品味以及帕金森定律方面的见解、为什么你需要痴迷于“让用户思考”、他那篇传奇备忘录《我们这里不卖马鞍》的背景故事等等。非常感谢 Noah Weiss, Chris Cordell, Ali Rael 和 Johnny Rogers 为本次对话提供的建议和问题。这是一期非常特别的内容，我真心希望以后能再请 Stewart 回来深入探讨。

[00:02:27] Lenny Rachitsky

English:

If you enjoy this podcast, don't forget to subscribe and follow it in your favorite podcasting app or YouTube. It helps tremendously. And if you become an annual subscriber of my newsletter, you get 17 incredible products for free for an entire year, including Devin, Lovable, Replit, Bolt, n8n, Linear, Superhuman, Descript, Wispr Flow, Gamma, Perplexity, Warp, Granola, Magic Patterns, Raycast, ChatPRD, and Mobbin. Head on over to lennysnewsletter.com and click Product Pass. With that, I bring you Stewart Butterfield after a short word from our sponsors.

中文翻译:

如果你喜欢这个播客，请不要忘记在您常用的播客应用或 YouTube 上订阅和关注。这对我们帮助很大。如果你成为我的时事通讯（Newsletter）的年度订阅者，你将免费获得 17 款不可思议的产品一整年的使用权，包括 Devin, Lovable, Replit, Bolt, n8n, Linear, Superhuman, Descript, Wispr Flow, Gamma, Perplexity, Warp, Granola, Magic Patterns, Raycast, ChatPRD 和 Mobbin。请访问 lennysnewsletter.com 并点击 Product Pass。在听完赞助商的简短介绍后，我们将为您带来 Stewart Butterfield 的访谈。

[00:02:57] Lenny Rachitsky (Sponsor: WorkOS)

English:

Here's a puzzle for you. What do OpenAI, Cursor, Perplexity, Vercel, Plaid and hundreds of other winning companies have in common? The answer is they're all powered by today's sponsor, WorkOS. If you're building software for enterprises, you've probably felt the pain of integrating single sign-on, SCIM, RBAC, audit logs, and other features required by big customers. WorkOS turns those deal blockers into drop-in APIs with a modern developer platform built specifically for B2B SaaS. Whether you're a seed-stage startup trying to land your first enterprise customer or a unicorn expanding globally, WorkOS is the fastest path to becoming enterprise-ready and unlocking growth. They're essentially Stripe for enterprise

features. Visit WorkOS.com to get started or just hit up their Slack support where they have real engineers in there who answer your questions superfast. WorkOS allows you to build like the best with delightful APIs, comprehensive docs, and a smooth developer experience. Go to workos.com to make your app enterprise-ready today.

中文翻译:

给你出一个谜题：OpenAI, Cursor, Perplexity, Vercel, Plaid 以及数百家其他成功的公司有什么共同点？答案是：它们都由今天的赞助商 WorkOS 提供支持。如果你正在开发企业软件，你可能感受过集成单点登录 (SSO)、SCIM、基于角色的访问控制 (RBAC)、审计日志以及大客户要求的其他功能的痛苦。WorkOS 通过专门为 B2B SaaS 构建的现代开发平台，将这些交易阻碍变成了即插即用的 API。无论你是试图赢得第一个企业客户的种子期初创公司，还是正在全球扩张的独角兽，WorkOS 都是实现企业级就绪并解锁增长的最快路径。他们本质上是企业级功能的“Stripe”。访问 WorkOS.com 开始使用，或者直接联系他们的 Slack 支持，那里有真正的工程师会飞速回答你的问题。WorkOS 让你能以愉悦的 API、详尽的文档和流畅的开发体验进行构建。立即访问 workos.com，让你的应用具备企业级能力。

[00:03:58] Lenny Rachitsky (Sponsor: Metronome)

English:

This episode is brought to you by Metronome. You just launched your new shiny AI product. The new pricing page looks awesome, but behind it, last-minute glue code, messy spreadsheets, and running ad hoc queries to figure out what to bill. Customers get invoices they can't understand. Engineers are chasing billing bugs. Finance can't close the books. With Metronome, you hand it all off to the real-time billing infrastructure that just works. Reliable, flexible, and built to grow with you. Metronome turns raw usage events into accurate invoices, gives customers bills they actually understand and keeps every team in sync in real-time. Whether you're launching usage-based pricing, managing enterprise contracts, or rolling out new AI services, Metronome does the heavy lifting so that you can focus on your product, not your billing. That's why some of the fastest growing companies in the world like OpenAI and Anthropic run their billing on Metronome. Visit metronome.com to learn more. That's metronome.com.

中文翻译:

本期节目由 Metronome 赞助。你刚刚发布了闪亮的 AI 新产品，定价页面看起来棒极了，但在后台，却是临时拼凑的代码、混乱的电子表格，以及为了弄清楚该收多少钱而运行的临时查询。客户收到看不懂的发票，工程师在修复计费漏洞，财务无法结账。有了 Metronome，你可以将这一切交给即时生效的实时计费基础设施。它可靠、灵活，且伴随你的业务成长。Metronome 将原始使用数据转化为准确的发票，为客户提供真正看得懂的账单，并让每个团队实时同步。无论你是要推出基于使用量的定价、管理企业合同，还是发布新的 AI 服务，Metronome 都能承担繁重的工作，让你专注于产品而非计费。这就是为什么像 OpenAI 和 Anthropic 这样全球增长最快的公司都在使用 Metronome。访问 metronome.com 了解更多信息。

[00:04:59] Lenny Rachitsky

English:

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

中文翻译:

Stewart，非常感谢你能来，欢迎来到本播客。

[00:05:02] Stewart Butterfield

English:

Thank you for having me. I'm excited.

中文翻译:

谢谢你的邀请。我很兴奋。

[00:05:05] Lenny Rachitsky

English:

I'm even more excited. I'm so honored to have you here. I never told you this, but you've been towards the very top of my wish list of guests I have on this podcast ever since I started this podcast a few years ago, so I'm very excited that we're finally making this happen. I have so many questions for you. My first question is just what the heck are you up to these days? I feel like ever since you left Slack, we haven't heard much from Stewart. I'm curious what you're up to you hopefully or just chilling.

中文翻译:

我比你更兴奋。能请到你真是我的荣幸。我从未告诉过你，自从几年前我创办这个播客以来，你一直是我最想邀请的嘉宾名单中的前几名，所以我非常激动我们终于实现了这次对话。我有太多的问题想问你。我的第一个问题是：你最近到底在忙些什么？感觉自从你离开 Slack 之后，我们就没怎么听到你的消息了。我很好奇你是在忙些什么，还是只是在享受闲暇时光。

[00:05:28] Stewart Butterfield

English:

I'm mostly just chilling. I left Salesforce two and a half years ago and I have a two and a half year old, so she was actually born three days after my last day, so a lot of time with family and it's an enormous privilege to be able to spend time with young kids while they're young. No new company to announce or anything like that. I do get a lot of emails and texts. Basically every three to six weeks there's this cycle because Cal Henderson who's the CTO of Slack and who also, we worked together on Flickr, so have worked together now for 23 years, have been talking about what we want to do next if there is something. But honestly, the big challenge has been I think these things are destroying the world and what we're good at is making software. So you find some way to make software that helped people use their phones less often, then that would be a big winner, but haven't come up with anything good. A lot of philanthropic work, nothing to announce there yet, but there's some cool projects that I'm working on, and a lot of just personal creative art projects and supporting other artists and stuff like that.

中文翻译:

我大部分时间都在放松。我两年前离开了 Salesforce，我还有一个两岁半的孩子，她实际上是在我离职三天后出生的。所以我有大量时间陪伴家人，能在孩子年幼时陪伴他们是一种巨大的特权。目前没有新公司要宣布。我确实收到很多邮件和短信，基本上每三到六周就会有一个周期，因为 Cal Henderson (Slack 的 CTO，我们也一起做过 Flickr，合作了 23 年) 一直在讨论如果有机会的话，我们下一步想做什么。但老实说，最大的挑战在于，我认为这些电子设备 (手机) 正在毁灭世界，而我们擅长的是做软件。如果你能找到某种方法做出能让人们减少手机使用频率的软件，那将是一个巨大的成功，但目前还没想到什么好点子。我做了很多慈善工作，目前还没什么可宣布的，但有一些很酷的项目正在进行中，还有很多个人的创意艺术项目，以及支持其他艺术家之类的事情。

[00:06:45] Lenny Rachitsky

English:

To prep for this chat, I talked to so many people that have worked with you over the years to try to figure out what you taught them about building product, building teams, building companies that most stuck with them, that most helped them build amazing products. The first is a concept called utility curves. This came up a bunch across so many people that have worked with you. Talk about what is a utility curve, how you use that to build better products.

中文翻译:

为了准备这次谈话，我采访了许多这些年来与你共事过的人，想弄清楚你教给他们的关于打造产品、团队和公司的知识中，哪些最让他们刻骨铭心，哪些最能帮助他们打造出惊艳的产品。第一个概念叫“效用曲线”（Utility Curves）。很多共事过的人都提到了这一点。请谈谈什么是效用曲线，以及你如何利用它来打造更好的产品。

[00:07:08] Stewart Butterfield

English:

This is pretty easy because it's a very familiar S-curve where you have, it's flat and it starts arcing up and then there's a really steep part and then it levels off again. And on the horizontal axis, you can think of cost or effort and on the vertical axis, it's value or convenience. It depends exactly what you're talking about, but the idea is the first bit of effort you put into something doesn't result in a huge amount of value. And then there's some magic threshold where it produces an enormous amount of value and then continued investment doesn't really pay off. The most basic example I can think of is let's say you're making a hammer, and on that bottom axis, it's now quality, and if the hammer has a handle that breaks with any impact, then is totally useless. And if you make it a little bit stronger, it's still pretty useless and it's like junk, junk, junk, junk, junk. Okay, good, great. Then it doesn't matter anymore. If you're making an app, okay, this app's going to have users and so let's make a user's table and a database, and so far you have generated no value. The reason I felt like this was so important is because we would talk about a feature, and usually features are thought of as a binary. You either have this feature or you don't. The argument I guess was have we just not invested enough in this or have we got all the value or convenience or quality or whatever that we could get out of this? And we had pointed diminishing returns and it just doesn't matter. I think in many cases, people will add a feature, it's not good enough and so people don't use it or appreciate it, but now you've added some complexity to the app and then people give up or take it back or they try something in testing and they don't get the results they want, and so they decide that this a thing is worth doing. We would try to really investigate and decide whether we were on the first shallow part of the curve, the second shallow part of the curve, or we're just coming up to it. So I think it's a lot easier to understand the value of this when you're talking about a specific app and a specific feature, but I think it was ultimately helpful in getting people to understand whether something was worth it or not.

中文翻译:

这其实很简单，因为它就是大家熟悉的 S 型曲线：开始是平缓的，然后开始向上弯曲，接着进入一个非常陡峭的部分，最后再次趋于平缓。在横轴上，你可以看作是成本或投入；在纵轴上，则是价值或便利性。这取决于你具体讨论的对象，但核心思想是：你最初投入的一点努力并不会产生巨大的价值。然后会有一个神奇的阈值，跨过它就会产生巨大的价值，而之后的持续投入收益就不再明显了。我能想到最基础的例子是做一把锤子：横轴是质量，如果锤柄一敲就断，那它完全没用。如果你把它做强一点点，它依然没用，就像一堆垃圾。直到它达到“好用”的程度，再往后增加质量就没那么重要了。如果你在做一个 App，你说我们要建一个用户表和数据库，到目前为止你还没创造任何价值。我之所以觉得这很重要，是因为我们讨论功能时，通常认为它

是二元的：要么有这个功能，要么没有。但我的观点是：我们是对这个功能投入不足，还是已经榨干了它能提供的所有价值、便利或质量？我们是否已经到了收益递减点，再做下去也没意义了？我认为在很多情况下，人们添加了一个功能，但做得不够好，导致用户不用也不领情，结果你只是增加了App的复杂度。然后人们可能会放弃这个功能，或者在测试中没拿到想要的结果，就觉得这事不值得做。我们会深入调查，判断我们是处于曲线起步的平缓期，还是快到爆发期，还是已经到了顶部的平缓期。当你针对具体的App和功能讨论时，这个概念会更容易理解，它最终能帮助人们理解某件事到底值不值得做。

[00:09:31] Lenny Rachitsky

English:

So just to mirror back what I'm hearing, there's this, if you visualize this curve at the bottom, it's like I don't even know what this is. And then up the curve is like, okay, I sort of get it. And then at the top is, okay, I can't live without this now that I understand what this is for, it feels like it's a really a different way of thinking about getting to the aha moment for someone where they see, okay, saved items, I get it, I need to use this constantly. It feels like this works both for a specific feature and also just for Slack, getting people to even understand here's what Slack can do for you. And then now I can't live without Slack. And essentially this is a lens you use to figure out where to spend product resources because if you don't get up that curve to I get it and I can't live without it, nothing else matters. Is that the framework?

中文翻译:

所以根据我的理解，如果你想象这条曲线，底部就像是“我甚至不知道这是什么”。随着曲线向上，变成了“好吧，我大概明白了”。到了顶部则是“既然我明白了这是干什么的，我现在离不开它了”。这似乎是一种思考如何带用户到达“Aha Moment”（顿悟时刻）的独特方式，让他们意识到“哦，收藏功能，我懂了，我得经常用”。这似乎既适用于特定功能，也适用于Slack整体——让人们理解Slack能为你做什么，直到他们离不开它。本质上，这是你用来决定产品资源投向的透镜，因为如果你不能让用户爬上那条曲线，达到“我懂了且离不开它”的程度，其他一切都不重要。是这个框架吗？

[00:10:13] Stewart Butterfield

English:

Yeah, and I think then you layer on another concept like the, Bezos used the term divine discontent. The line actually moves because once people are familiar with a piece of software or the way a feature is implemented or something like that, their standards go up, and so there's this competition. And again, this axis can be, utility is the best general term for it, but it could be quality, convenience, speed, it could be any number of things, but as you improve your search capability or as you improve your login experience or your forget password experience or your checkout experience or whatever everyone else is as well. And so there's this continued investment and when forget about thinking about a new feature, you're looking at how the product works overall and usually things get implemented once, and then if they're lucky, they get improved upon periodically. Most things get improved upon very infrequently and some things get improved upon never. I want to give an example at the absolute extreme because I actually don't know how long this has been, but I try not to criticize other people's software so much because I'm very familiar with the trade-offs and prioritization and how hard it can be and blah, blah, blah, blah. But okay, so most people have the Gmail Calendar app on their phone. I travel a fair bit. I'm mostly in the Eastern Time Zone, sometimes in Mountain Time, sometimes in Pacific, sometimes in English time, and sometimes in Japan, Central Europe. There's maybe 10 time zones, 12 time zones that I would ever choose. When you hit the option to set the time zone on an event in Google Calendar, on the iOS app, it presents all the time zones in the world in alphabetical order. And I mean, there's probably worse orderings, but there's no value in that. And even when you start searching, it still presents them in

alphabetical order by country with that turn. So if I'm in California and I'm trying to set the appointment for next week when I'm back in New York and I type in E-A-S-T and I get a bunch of garbage, okay, Eastern, and then the first one is Eastern Australia, New South Wales, and then Eastern Australia, Queensland, and then Eastern Australia, Daylight Savings and Eastern Australia standard time. And then you're like, "Well, fuck, I can't remember which one is Daylight Savings and which one is standard time?" I could keep going like this for a while. This is an app that's used by at least hundreds of millions of people, presumably every single Google employee. It's bananas how bad it is. There's so many, there's all these clever things you could do. Like you know me, I'm on the West Coast, first option should be the East Coast and vice versa. But it definitely shouldn't be that every time zone is presented with equal value. I don't know a couple hundred time zones. I grew up in Canada. Newfoundland has its own time zone, which is offset by half an hour. The population of Newfoundland has about half a million people. Not that many people go to visit Newfoundland, maybe a million people in all of history so like a million and a half out of 8 billion people. And there's Newfoundland, the same with China time, which is like 25% of the world's population in this. Anyway, that was a little bit longer than I intended to go on this example, but it's crazy because no one's going to switch to Gmail or to G Suite, Google Calendar from Outlook Exchange because the time zone picker is good, so maybe in some sense it doesn't matter, but at the same time there's a real value in delighting customers and there's an emotional connection that they form or don't form. And in some cases that could be really positive like they would recommend it. And when they switch companies or decide to start their own company, they're going to choose to use this product or advocate for it because of that emotional connection and vice versa. They'll also be like, "I hate this thing that drives me bananas. I really think we should stop using it," or advocate for the alternative. And I think people just don't appreciate or come back to those things often enough. And then there's this category of really essential parts of the app again like account creation, sign up, forgot password, things like that, that for most organizations very infrequently get a lot of love and iteration and improvement despite the fact that the quality bar has gone up across the board and continually goes up.

中文翻译:

是的，而且我认为你还要叠加另一个概念，就像贝佐斯用的那个词——“神圣的不满”（Divine Discontent）。那条基准线实际上是在移动的，因为一旦人们熟悉了某个软件或某个功能的实现方式，他们的标准就会提高，竞争也就随之而来。同样，纵轴可以用“效用”来概括，但也可以是质量、便利性、速度等等。当你改进搜索能力、登录体验、找回密码体验或结账流程时，其他人也在改进。所以这需要持续的投入。先别想新功能，看看产品整体是如何运作的。通常功能只会被实现一次，运气好的话会定期改进，但大多数东西改进频率极低，有些甚至从未改进。我想举一个极端的例子，虽然我尽量不批评别人的软件，因为我很清楚权衡和优先级有多难。但是，大多数人的手机上都有 Gmail 日历。我经常出差，有时在东部时间，有时在山地时间，有时在太平洋时间，有时在英国、日本或中欧。我可能只会用到 10 到 12 个时区。当你在 iOS 版 Google 日历上设置事件时区时，它按字母顺序列出了世界上所有的时区。虽然可能有更糟的排序方式，但这毫无价值。甚至当你开始搜索时，它依然按国家字母顺序排列。如果我在加州，想给下周回纽约后的行程设个时间，我输入“EAST”，结果出来一堆垃圾：首先是东澳大利亚新南威尔士，然后是昆士兰，然后是夏令时，然后是标准时。你会想：“妈的，我哪记得哪个是夏令时哪个是标准时？”我可以一直吐槽下去。这是一个至少有数亿人使用的 App，大概每个 Google 员工都在用，但它做得这么烂简直不可理喻。有很多聪明的方法可以改进，比如你知道我在西海岸，第一个选项就应该是东海岸，反之亦然。绝对不应该把每个时区都赋予同等的权重。我不需要几百个时区。我在加拿大长大，纽芬兰有自己的时区，偏移半小时。纽芬兰只有 50 万人口，历史上可能只有 100 万人去过那里，也就是 80 亿人里的 150 万。但纽芬兰时区就在那，和占世界人口 25% 的中国时间权重一样。抱歉说这个例子说长了，但这很疯狂，因为没人会因为时区选择器做得好就从 Outlook 换到 Gmail。所以从某种意义上说，这可能不重要，但同时，让客户感到愉悦是有真正价值的，这关乎他们是否能与产品建立情感连接。如果有这种连接，他们会推荐它；当他们换公司或创业时，会因为这种情感连接而选择或拥护这个产品。反之亦然，他们会说：“我讨厌这玩意，它让我抓狂，我们别用了。”我认为人们对这些细节的重视程度不够。还有一类非常核心的部分，比如账号创建、注册、找回密码，尽管整个行业的质量标准在不断提高，但大多数组织很少对这些部分投入精力和迭代。

[00:15:11] Lenny Rachitsky

English:

Let's go down that rabbit hole a little bit more around delight and craft. Slack was famous for being one of the early, let's say consumerized B2B SaaS products. Slack leaned into delight and experience and craft and a great experience. And you just as a product leader, I'd say are known as very taste forward, very craft oriented leader, which is pretty rare and I think continues to be rare. So there's a few things I want to talk about here. One is taste. I heard at a talk, you gave a talk on taste and you have a really unique perspective on just what taste is, what product taste looks like. Can you share that?

中文翻译:

让我们再深入探讨一下关于“愉悦感”和“工艺”的话题。Slack 以早期“消费级”B2B SaaS 产品著称，它非常注重愉悦感、体验和工艺。作为一名产品领导者，你以“审美领先”和“注重工艺”而闻名，这在当时很罕见，现在依然罕见。我想聊几件事，首先是“品味”(Taste)。我听过你关于品味的演讲，你对什么是品味、产品品味长什么样有非常独特的见解。能分享一下吗？

[00:15:49] Stewart Butterfield

English:

There is a lot of you going back to the utility curves again, people who are obsessed with this one little thing and keep on adding more and more detailed improvements beyond the point where it makes much of a difference. But I guess a couple of things about taste. So one is can you learn to develop it? I think so because the word literally comes from experiencing food and putting stuff in your mouth. And can people become better chefs with training? Yes, absolutely. Undoubtedly, some people have a natural advantage and are born with this ability to make discernments that are difficult for other people to make and stuff like that. But you can definitely practice and you can definitely get better. The second thing I'd say is you can create a real advantage for yourself, for your product, for your company by leaning into it because most people don't have good taste and don't invest. You're probably familiar with, again, Jeff Bezos line, your margin is my opportunity and pretty obvious what he meant by that. I would tell the story at Slack over and over again. It actually made it part of the new hire welcome. I'm in Vancouver at our Vancouver office and I'm going for a walk with Brandon Velestuk who's our, at the time creative director for product development, I think that was his title. And we're in the Yaletown neighborhood in Vancouver so there's really narrow sidewalks because it used to be a warehouse district and now it's fancy restaurants and nail salons and boutiques and stuff. And as it does in Vancouver, it starts to rain. We don't have umbrellas. We're walking back to the office and most people have umbrellas and we're on these narrow sidewalks with people coming towards us with umbrellas. We noticed how few people would move their umbrella out of the way. And of course, the other person, their umbrella, the pokey bits are exactly at eye level for people walking towards them. We would get forced off the sidewalk or having to duck down or whatever. It became a game like we were guessing is this person going to tilt their umbrella out of the way so we can pass or not? And something like one-third of the people would do it. And we had this conversation about it where it's like, okay, I can think of three reasons why people wouldn't do it. One is they have very few avenues in their life to exercise power and this is one of them. And they're just, want to get out there and dominate people and cause suffering. Shouldn't ascribe to malice that which can be ascribed to ignorance so that probably is the explanation for a tiny, tiny, tiny percentage of people. But the other two explanations aren't that great either. One is that they see it's happening, they see they're pushing other people off the sidewalk or poking them in the eye or whatever, and they're just like, "Fuck, that's too bad. I wish there was something I could do about that, but I can't think of anything." And the last reason is

they just don't notice it all. They're just oblivious to their impact on other people. And they're so in their head, and I can't really think of any other explanations for it besides that. And so we would say it's not like tilting your umbrella is our opportunity. That's not a great rephrase of your margin is my opportunity, but your failure to really be consider it exercise this courtesy and really be empathetic about other people's experience is an advantage that you can create a critical advantage. I think that there's many reasons why Slack was successful at the moment. It was successful and we think we had a bunch of really wonderful tailwinds and all of that stuff, but it wouldn't have grown the way it did without those little conveniences which caused people to form an emotional connection because a lot of our growth came from startup A uses Slack, and then someone leaves startup A for startup B, and startup B doesn't use Slack yet. And they would be like, "Oh my God, you guys, you really, this is so good. We got to try it." And the spread was driven by that and people really genuinely advocating for it.

中文翻译:

回到效用曲线，有些人痴迷于某个小细节，不断添加细微的改进，甚至超过了产生实际影响的临界点。关于品味，我想说两点：第一，品味可以培养吗？我认为可以，因为这个词字面上源于品尝食物。通过训练，人们能成为更好的厨师吗？绝对可以。无疑，有些人天生就有优势，能做出别人难以察觉的细微分辨，但你绝对可以通过练习变得更好。第二，通过倾注于品味，你可以为自己、产品和公司创造真正的优势，因为大多数人没有好品味，也不愿投入。你可能熟悉杰夫·贝佐斯那句话：“你的利润空间就是我的机会。”他什么意思很明显。我在 Slack 反复讲一个故事，甚至把它变成了新员工入职培训的一部分。当时我在温哥华办公室，和当时的创意总监 Brandon Velestuk 一起散步。我们在耶鲁镇，那里的街道很窄，以前是仓库区，现在全是高档餐厅和精品店。温哥华下起了雨，我们没带伞。窄窄的人行道上，迎面走来的人都撑着伞。我们注意到，很少有人会主动把伞移开。对方伞尖的高度正好在迎面走来的人的眼睛水平线上。我们不得不被挤下人行道，或者低头躲避。这变成了一个游戏：我们猜测这个人会不会倾斜雨伞让我们过去？结果只有大约三分之一的人会这么做。我们讨论了原因，我想到了三个理由：第一，他们在生活中几乎没有行使权力的机会，这是其中之一，他们只想支配别人，制造痛苦。但“不要把可以用无知解释的事情归咎于恶意”，所以这可能只解释了极小一部分人。另外两个解释也不怎么样：一是他们看到了正在发生的事，看到自己把别人挤下路或戳到别人眼睛，但他们想：“妈的，真糟糕，我希望我能做点什么，但我实在想不出办法。”最后一个原因是他们根本没注意到，完全沉浸自己的世界里，对给别人造成的影响视而不见。所以我们会说，虽然“倾斜你的雨伞是我的机会”听起来不像“你的利润是我的机会”那么响亮，但你未能体贴他人、未能行使这种礼貌、未能对他人的体验产生共情，这就是我们可以创造的关键优势。Slack 成功有很多原因，比如顺风顺水的时机，但如果没那些让人们产生情感连接的小便利，它不会增长得那么快。我们的增长很多来自于：初创公司 A 用 Slack，有人离职去了初创公司 B，而 B 还没用 Slack。他们会说：“天哪，伙计们，这东西太好用了，我们必须试试。”这种传播是由人们发自内心的拥护驱动的。

[00:20:07] Lenny Rachitsky

English:

That is an amazing metaphor. I love that one moment became a value of product craftsmanship at Slack.

中文翻译:

这是一个很棒的比喻。我喜欢这个瞬间变成 Slack 产品工艺价值观的过程。

[00:20:14] Stewart Butterfield

English:

Tilt your umbrella was a very common saying on company swag and stuff like that.

中文翻译:

“倾斜你的雨伞”成了公司周边产品上非常流行的一句话。

[00:20:20] Lenny Rachitsky

English:

Is there an example, I imagine there are many, but from the time of building Slack, especially in the early days where you chose to go big on craftsmanship and experience and delight versus speed where you thought looking back that was a really great idea and worth really core just to success.

中文翻译:

有没有一个例子（我想肯定有很多），是在 Slack 早期，你选择在工艺、体验和愉悦感上大举投入，而不是单纯追求速度，回过头看你觉得那是一个非常棒的主意，甚至是成功的核心？

[00:20:37] Stewart Butterfield

English:

Here's a bunch of little examples. Someone else came up with this idea, and I'm trying to remember who it was, but let's see, maybe Andrea Torres, maybe Ben Brown, something like that who was like, "Why did we ask people for email address and password if their ownership of the email address was the thing that allowed them to create the account in the first place? Why don't we just ask them for their email address and then send them a link?" And so when Slack's first version of the mobile app came out, we're like, "Typing your password on your phone if you have any minimal threshold of password hygiene is a terrible experience." Capital H, lowercase Q, six, caret, period. So let's just have them enter the email address. We'll send them a link. The link will automatically open the app and authenticate them. And so there's one, a little example.

中文翻译:

这里有几个小例子。有人提出了这个想法（我不记得是谁了，可能是 Andrea Torres 或 Ben Brown），他说：“既然拥有邮箱地址本身就是创建账号的前提，为什么我们还要问用户要邮箱和密码呢？为什么不直接问他们要邮箱，然后发给他们一个链接？”所以当 Slack 的第一个移动端版本发布时，我们认为：“在手机上输入密码（如果你有起码的密码安全意识，密码会很复杂）简直是噩梦。”大写 H，小写 Q，数字 6，脱字符，句号……太痛苦了。所以我们就让他们输入邮箱，我们发个链接，点击链接自动打开 App 并完成认证。这就是一个小例子。

[00:21:33] Lenny Rachitsky

English:

Wow. So you guys invented the magic link experience.

中文翻译:

哇，所以是你们发明了“魔术链接”（Magic Link）的体验。

[00:21:36] Stewart Butterfield

English:

Someone else invented. I want to be clear that I had seen that idea somewhere else, someone else, a blog post about it or something like that. But we were the first ones, to my knowledge, that really scaled that and made it a standard. There is another one which we really puzzled about in the very early days where people have a long history of using messaging apps from AOL Instant Messenger to SMS to WhatsApp, where their expectation is they get a notification for every message that's received. And in the case of Slack, that doesn't make as much sense because you're a member of many channels and the messages may not be for you, and so that's why we have the @ tagging people. And we certainly didn't invent that, that was Twitter. But what we realized was people were signing up for Slack, and it's one engineer on this team inside of this larger organization, inside this larger company, and they would pull in the person next to them and they would say, "Let's try it out." And then they would send a message and then one person would be like, "I didn't get a notification. This is bullshit." We reluctantly decided that we had to send notifications for every single message as the default for new accounts. But once you had, I don't remember what the threshold would happen, I think it's once you had received 10 messages, we would pop up this little thing that says, "Hey, you have our default settings for notifications. We don't want Slack to be noisy for you. Would you like to switch to our recommended settings?" And then they would just click a link and it would have what should be the default, which is, you only get a notification if it's a DM or someone tags you. But we realized it was worth that investment to get people over the hump. I'll give one more simple one and then one kind of more complex. One, people would just like the, I can't remember if it's called urgent or important, but the flag in Outlook, set the priority of a message for the recipients always got abused inside of every company. As soon as someone does it, everyone's like, "Okay, I'm going to do that too for my message." And so all of your messages have the little flag and it becomes useless. We have @everyone, which causes a notification to be sent to every member of the channel when the message is sent and people would start, someone would find this feature inside of a organization. They would @everyone, everyone would get a notification and then the next person to send a message who was like, " Well, my thing's more important than Bob's thing. I'm going to also @everyone." And it became really obnoxious and people would complain about it, but it was, I don't know, I guess tragedy of the commons. It's not quite exactly the same thing, but it was this real dynamic that happened over and over again. So we came up with what was called the shouty rooster, and internally we said, "Don't be a cock." But we didn't obviously say that publicly when you @everyone, a little rooster would pop up and it would have you sound waves coming out of its mouth and being really obnoxious and say, "Hey, this is going to cause a notification for 147 people in eight different time zones. Are you sure you want to send this message with the @everyone?" And of course, that worked amazingly and it dropped off. And again, it was really trying to shape people's behavior so that they used, one is not to be very flexible, but we knew that there was ways to use it that would be annoying and difficult for everyone. And so try to shape the communication culture inside the organization to take best advantage on it.

中文翻译:

是别人发明的。我要澄清一下，我在别处见过这个点子，比如某篇博客。但据我所知，我们是第一个真正大规模应用并使其成为标准的公司。还有一个我们在早期非常纠结的问题：人们长期使用 AOL、短信、WhatsApp，习惯了每收到一条消息就有一个通知。但在 Slack 里，这没意义，因为你加入了很多频道，消息可能不是发给你的，所以我们有了 @ 功能（这当然是 Twitter 发明的）。但我们发现，当人们注册 Slack 时（比如大公司里的一个小团队），他们拉来旁边的同事说“试试看”，发了一条消息，对方却说：“我没收到通知，这玩意真烂。”于是我们无奈地决定，新账号默认对每条消息都发通知。但一旦你收到了（我不记得具体阈值了，大概是 10 条）消息，我们会弹出一个小提示：“嘿，你现在用的是默认通知设置。我们不想让 Slack 吵到你，你想切换到我们的推荐设置吗？”点击链接后，就会变成真正的默认设置：只有私聊或被 @ 时才通知。我们意识到，为了让用户跨过那个门槛，这种投入是值得的。再举一个简单和复杂的例子。在 Outlook 里，人们总爱滥用“紧急”或“重要”标志。一旦有人用了，其他人也会跟着用，最后所有消息都带个小红旗，标志就失效了。Slack 有 @everyone 功能，发消息时全频道都会收到通知。有人发现这招好使，就开始 @everyone，接着下一个人

觉得自己的事比 Bob 的更重要，也跟着 @everyone。这变得非常讨人厌，就像“公地悲剧”。于是我们设计了“尖叫的公鸡”(Shouty Rooster)，内部我们叫它“别当混蛋”(Don't be a cock)。当你 @everyone 时，会弹出一只尖叫的公鸡，显示声波，非常显眼地提醒你：“嘿，这会给 8 个时区的 147 个人发通知，你确定要这么做吗？”这招效果极佳，滥用行为立刻减少了。我们是在尝试塑造人们的行为，引导组织内部的沟通文化，从而发挥产品的最大优势。

[00:25:02] Lenny Rachitsky

English:

That feature still exists. I see that rooster all the, no, I don't see it all, well, actually I do @channel, because I run a big Slack, so I see that rooster, that survived.

中文翻译:

那个功能还在。我经常看到那只公鸡……不，也不是经常，但我管理着一个很大的 Slack 社区，当我用 @channel 时，我能看到那只公鸡。它活下来了。

[00:25:11] Stewart Butterfield

English:

Yeah. Yeah, that survived and good because it was a trivially easy thing to implement and made a really big difference. But it also taught people how the product worked, because people probably didn't know that @everyone or @channel... Didn't think about the cost, at least.

中文翻译:

是的，它活下来了，这很好，因为它实现起来非常简单，却能产生巨大的影响。它还教会了人们产品是如何运作的，因为人们可能之前并不知道 @everyone 或 @channel 的代价。

[00:25:31] Lenny Rachitsky

English:

Genius.

中文翻译:

天才的设计。

[00:25:32] Stewart Butterfield

English:

Yeah. Here's one more. So we decided we were going to Do Not Disturb as a feature. And we had this, not conundrum, but you're trying to take into account all the different uses of Slack because by the time we implemented this, 2017, there was tens of thousands of paying customers, the organizations, hundreds of probably millions of users, maybe hundreds of thousands of organizations. I don't remember how many. And everyone had set up stuff the way that they liked it, including things like ops alerts going into channels for on-call engineers for some of the biggest systems and apps in the world. And so we couldn't just deploy it right away. We realized that some of the decision-makers, the owners of the organizations were going to have really strong opinions about this. We also realized that some of the end users are

going to have strong opinions, and we wanted to figure out a way to balance the concerns and give people appropriate means of control. So we came up with this really elaborate system for the rollout, which was, we told everyone, I'm sorry, every Slack administrator that this was coming weeks before it came. And we told them that we were going to set a default for their organization, which I believe was either 7:00 P.M. to 7:00 A.M. in their local time zone, or 8:00 A.M. to 8:00 P.M., I can't remember which it was, but also that they could override that default, and also that the individual end users could override that system owner default. And finally, that the system owner could, if they changed the default again, would override all of the end user's preferences and then the end users could override them again. And it wasn't to create this dynamic where people were at war, but so that you could change a policy and then people could still customize and stuff like that. But this was a much longer and more convoluted process, but it allowed the millions of people who were using Slack to get the feature without creating a bunch of conflict and without people turning it off automatically. And I think critically, with setting a bunch of defaults, because if we didn't set the default, most people wouldn't turn it on at all. If we didn't default you to Do Not Disturb from 8:00 P.M. to 8:00 A.M. you probably, if you're the average person, wouldn't ever do it yourself. So that's another elaborate example where I think that investment made sense because it was a critical feature for a lot of people. And if we hadn't done it that way, I think it would've caused a lot of complaints and conflict and stuff like that.

中文翻译:

还有一个例子。我们决定要做“勿扰模式”(Do Not Disturb)。我们面临一个难题：到2017年我们实现这个功能时，已经有数万家付费机构和数百万用户。每个人都按自己的喜好设置好了，包括一些全球最大系统的运维警报。我们不能直接上线。我们意识到，组织的所有者(决策者)会有强烈意见，终端用户也会有强烈意见。我们得平衡各方顾虑。于是我们设计了一个非常复杂的发布流程：在上线前几周通知所有管理员，告诉他们我们将为组织设置一个默认值(比如当地时间晚7点到早7点)。管理员可以覆盖这个默认值，个人用户也可以覆盖管理员的设置。最后，如果管理员再次更改默认值，它会覆盖所有用户的偏好，然后用户可以再次手动覆盖。这并不是为了让大家“开战”，而是为了让政策变更和个人定制共存。虽然过程漫长且复杂，但它让数百万用户在没有冲突的情况下获得了这个功能。关键在于设置默认值，因为如果不设默认，大多数人根本不会去开启它。如果你是普通用户，我们不帮你默认开启晚8点到早8点的勿扰，你可能永远不会自己去设。这是一个为了关键功能而进行复杂投入的例子，如果没这么做，可能会引发大量投诉和冲突。

[00:28:22] Lenny Rachitsky

English:

Those are amazing examples. I very much appreciate that Do Not Disturb feature when you guys launched that. I still remember that coming out. I'm sure a lot of people are very thankful for that.

中文翻译:

这些例子太棒了。我非常感激你们推出的勿扰模式，我至今还记得它发布的时候。我相信很多人都心存感激。

[00:28:30] Stewart Butterfield

English:

Yeah.

中文翻译:

是的。

[00:28:31] Lenny Rachitsky

English:

Something else I heard that you often espouse, which is counterintuitive to a lot of people is about friction, friction in the product experience. That friction is actually often a good thing. It's a feature, not a bug a lot of times if you use it well. Talk about your experience there.

中文翻译:

我还听说你经常拥护一个对很多人来说违反直觉的观点：关于产品体验中的“摩擦力”。你认为摩擦力往往是一件好事，如果运用得当，它是一个功能而非缺陷。谈谈你在这方面的经验。

[00:28:46] Stewart Butterfield

English:

Yeah. So yes, and there's also another issue around friction, which is it became like a mantra or just kind of an assumption that you should always be trying to remove friction. And in some cases that's true. We would talk about it in Slack. It was hard to market. It was hard to explain what it was if you had never used it before. You could say a messaging app for businesses or whatever, but a critical disadvantage to Slack doing out-of-home advertising, putting up a billboard versus beer or cars is, no one needs to be explained why they would want a car or beer, but everyone will have to explain one day why they want Slack. And so the problem there is comprehension, and this will come up an enormous amount. So now imagine you want to get tickets to the Taylor Swift concert in San Francisco and you go to the Ticketmaster website. If you think about both your comprehension, it's perfect to this case. And that translates into the specificity of your intent and the degree of your intent is also kind of maxed out. So look, I really want to get these tickets. I know exactly what they are. They're Taylor Swift tickets for this date at this venue. And so in that scenario, it doesn't really matter if Ticketmaster's website is slow, it doesn't really matter if the payments page errors out, you're going to persist and get through it. So obviously they're better to reduce friction, but in some sense there's not a huge amount of value in doing that. For most creators of products, there are a handful of cases where that really is true for you as well. And they include things like user registration, authentication, checkout flows for e-commerce. I am significantly more likely to buy something if there's Apple Pay or Shop Pay or something like that. I'm significantly less likely to carry through the purchase of something if I have to manually enter all of the fields of my address one at a time rather than having one of those address pickers. It's crazy, but the issue is my intent isn't always 100%, and the specificity of my intent isn't always 100%. So if your thing is direct to consumer T-shirts and you acquire customers through Instagram ads, all of them know what T-shirts are. It's like, "This looks like a good T-shirt to me." But I'm rarely 100% intent. I might have a very specific intent, but my intent's like 70%. So if you're, the amount of friction is above that, I'm not going to do it. But now, okay, people coming to Slack.com, some friend had mentioned Slack and talked their ear off at some point months ago, and then they saw a news article and then they saw someone's tweet and then they saw an ad on about the website they were visiting and they finally said, "Okay, I'm going to go to this website." So their intent is at the absolute minimum threshold, it was before that last event happened, they were below and now they're above, but they're just above. The specificity of their intent like, "I need to get Taylor Swift concerts for this date at this venue." Is also very low, because they're like, "It's a work thing. I'm not sure it's a spreadsheet or a calendar or exactly what it is." So they were coming in at 0.1% over these critical thresholds. What was the challenge? It wasn't friction, because it's not like they were aiming for something and they knew what they were aiming for and they were just trying to get themselves to that point. What we had to worry about was creating comprehension and in two senses, what is this thing? And what am I supposed to do next? And that creation of comprehension in the sense

of explaining stuff, that creation of comprehension in the sense of the design of the UI, of the screen, of the page or whatever, and the visual hierarchy and the affordances that are there and the indication of things to interact with and which thing should be the next thing to do and all that stuff, that becomes really critical. And I think very, very few people recognize that. They're like, "I want to get people who come to my webpage to the sign up form as quickly as possible." But if they don't know what they're signing up for and they don't know what it's going to do after, is it going to spam them? They don't know, "Am I going to have to pay on the next step or what?" Then they're just going to back out. And this was a lifelong battle because the remove friction orientation is so deep in people. Again, it really makes a difference in those cases where people do have an intent and they do know what they're trying to do is a poor approach when the challenge is really comprehension, and I think the secret is most, 70%, 80% or whatever of a product design is in that comprehension step because people, if they do ever open the preferences tab and look at all the options, rarely have an idea. And if you can't teach them or make it possible for them to discover what the capabilities are, then they're not going to take advantage of them and they're not going to get as much out of it. And I think that the trick is for most of the unique parts of any application, most of the specific things that your app, your product, your software does are areas where the challenge is going to be comprehension inside of friction. It really could be anything Shopify, the purpose of the service for its end users is generally going to be kind to clear. But most people, most first-time store openers don't know that they can get reports or if they know that they can get reports, they don't know what kinds of reports. And if they know what kinds of reports they can get, they don't know how they can tweak them and what the timing should be and which things that are more important to display. And I could go on and on and on and on, and people just don't recognize that. So I want to see if this is still true. I'm just going to open my phone and clock app. And they had the craziest description for alarms. It's a little bit different, but people can look at their own phone. So I have, it says alarms and it says sleep and a vertical bar, wake up and says, no alarm, and a button that says change. And then if you hit it, it says sleep is off. In order to automatically turn on sleep features and edit your schedule, you need to turn sleep on. So obviously sleep was a good name for this thing if you already had a way of getting people to understand it. If you don't, it's ungrammatical and incomprehensible and why would you ever do it? And I got to guess, it's been like this for years, 90 plus percent and maybe 98% of people just do what I do, which is that you just create, "I want the alarm on and I'm going to set the time for it." And I don't know what turning sleep on does, but it's just the lack of comprehension prevents people from getting the value. And I'm sure that there's a bunch of value behind turning sleep on, whatever that means and people spend a lot of time on those features and it integrates with biometrics and your watch or who knows. Again, I still don't know because turning sleep on is like, what does that do? And what is it going to cost me? And what impact it's going to have? Those examples are just to me all over the place. And the reason I don't use most software where there was an actual choice point or the reason I don't use most features where there was a choice point for me is because I didn't understand what they were going to do and I don't give a shit. And if there is one mantra that I would use to replace that it's, Don't Make Me Think, I don't know if you remember that book.

中文翻译:

是的。关于摩擦力，人们已经形成了一种思维定式，认为应该始终尝试消除它。在某些情况下确实如此。在 Slack，我们很难向没用过的人解释它是什么。你可以说它是“企业级聊天软件”，但 Slack 做户外广告（比如广告牌）时，比起啤酒或汽车有一个致命劣势：没人需要被解释为什么要买车或喝啤酒，但每个人都需要被解释为什么要用 Slack。所以这里的问题在于“理解力”（Comprehension）。想象一下你想买泰勒·斯威夫特旧金山演唱会的门票，你去 Ticketmaster 网站。在这种情况下，你的理解力是完美的，你的意图非常明确且强烈：我就是要买这张票。在这种场景下，网站慢一点、支付页面报错都没关系，你会坚持下去。所以虽然减少摩擦更好，但其实没那么关键。对于大多数产品创造者来说，只有少数情况“减少摩擦”是核心，比如注册、认证、电商结账。如果有 Apple Pay，我购买的可能性会大增；如果让我手动输入地址，我可能就不买了。但问题是，用户的意图并不总是 100% 明确的。如果你在 Instagram 上卖 T 恤，大家都知道 T 恤是什么，但我的购

买意图可能只有 70%。如果摩擦力超过了这个值，我就不买了。但对于 Slack.com 的访客，他们可能几个月前听朋友提过，看过新闻或推特，最后才点进来。他们的意图刚过及格线，而且意图非常模糊：“这是个办公软件，但不确定是表格、日历还是什么。”他们是带着 0.1% 的微弱兴趣进来的。此时的挑战不是摩擦力，而是建立“理解力”：这到底是什么？我下一步该做什么？这种理解力包括文字解释、UI 设计、视觉层级、示能（Affordances，指界面元素的引导性）等。很少有人意识到这一点。他们只想让用户尽快到达注册表单，但如果用户不知道注册后会发生什么、会不会被发垃圾邮件、下一步要不要付钱，他们就会退出。这是一场持久战，因为“消除摩擦”的观念根深蒂固。当挑战在于理解力时，消除摩擦是错误的方法。产品设计的 70%-80% 应该花在“理解力”这一步。如果你不能让用户发现并理解产品的功能，他们就不会使用。比如 Shopify，虽然目的很明确，但很多新手店主不知道可以看报表，或者不知道报表可以怎么调。我再举个例子，打开手机时钟 App，它对闹钟的描述很奇怪。它写着“睡眠 | 起床”，显示“无闹钟”，有个“更改”按钮。点进去说“睡眠已关闭。为了自动开启睡眠功能并编辑日程，你需要开启睡眠。”如果用户已经理解了，那“睡眠”是个好名字；如果不理解，这就是不合语法且难以理解的。我敢打赌 98% 的人像我一样，只是设个闹钟，根本不去碰“开启睡眠”，因为不理解它能带来什么价值，也不想去思考。如果非要用一个口号来取代“减少摩擦”，那就是《别让我思考》（Don't Make Me Think）。

[00:37:02] Lenny Rachitsky

English:

Absolutely.

中文翻译:

没错。

[00:37:04] Stewart Butterfield

English:

Yeah. And honestly, it's been many more than 10 years since I read it, so I don't even remember all of the examples in the book, but as a mantra that was up there with utility curves because for two reasons. One is it's just like it's expensive to make a decision. You literally burn glucose. There's a metabolic action. There's ATP created in the mitochondria and your neurons and a bunch of stuff is happening and people do get decision fatigue and there is cognitive cost of all these things. But also there's an emotional aspect, which is, if your software stops me a second and asks me to make a decision and I don't really understand it, you make me feel stupid. I'm like, "I don't understand this." Some people, maybe their orientation is, "Okay, the software is stupid." But I think most people are like, "Oh, I'm dumb." And if you ever talk to people who aren't especially technologically savvy, the canonical example is people who are under 50 talking to their parents about using some piece of software and what they're supposed to do, the parents always feel stupid like they're the ones that are wrong. And so if you're causing people to think, in the best case, it's unnecessary use of their biological resources, and in the worst case you've now made them feel bad, emotionally bad, and they're going to associate that with the product forever. And these are things that are just kind of rolling one into the other. So I'm going to keep going with one last thing, because they just kind of come together, which is along with reduced friction, it's like reduce the number of clicks or taps it takes for someone to accomplish something which is almost always exactly the wrong thing. It's the easiest way you could make any action in your app, a single click or tap by just exposing every single possibility on one screen that scrolls for thousands and thousands and thousands and thousands of pages. And obviously that's terrible. So why do people think that a little bit of that is good? And here's an example. You open a menu, there's 14 things that people might want to do. Level one is group them into like items and put a vertical, sorry, horizontal divider between them so at least people can kind of chunk and see what there is. Step two is present the two or three most common things

or the five most common things, whatever and then have some form of other and then you go to a submenu that has more items and the decision of how to tune that becomes incredibly important. I'm going to pick on Google again just because it is, I feel like I'm Donald Trump here, but I'm going to interrupt myself again with a story. It's-

中文翻译:

虽然我读那本书已经是十多年前的事了，不记得所有例子，但这个口号和“效用曲线”一样重要。原因有二：第一，做决定是很昂贵的。你真的在消耗葡萄糖，这是一种代谢行为，线粒体在产生 ATP，神经元在放电，人们会产生决策疲劳，这些都有认知成本。第二是情感层面：如果你的软件停下来让我做决定，而我不理解，你就会让我觉得自己很蠢。有些人可能会觉得“这软件真笨”，但大多数人会觉得“哦，是我太笨了”。典型的例子是 50 岁以下的人教父母用软件，父母总觉得自己是错的，觉得自己笨。所以如果你让用户思考，最好的情况是浪费了他们的生理资源，最坏的情况是让他们产生了负面情绪，并永远将这种负面情绪与你的产品联系在一起。这些事情是环环相扣的。我再说最后一点，这和“减少摩擦”相关：人们总想减少点击或跳转次数，但这通常是错误的。如果你想让所有操作都一键完成，最简单的办法就是把所有选项都铺在一个屏幕上，让你滚上几千页，那显然很糟糕。为什么人们觉得减少点击就是好？举个例子：你打开一个菜单，有 14 个选项。第一级是把它们分组，加个分割线，让人们能分块查看。第二级是只展示最常用的 2 到 5 个，剩下的放进“其他”子菜单。如何调整这个比例至关重要。我要再次拿 Google 开刀了，虽然我觉得自己像在攻击别人，但我得讲个故事。

[00:39:57] Lenny Rachitsky

English:

Yeah, let's do it.

中文翻译:

好啊，讲讲看。

[00:39:58] Stewart Butterfield

English:

At some conference or event, I don't remember what it was, and this is probably eight years ago and we're in the bar after the sessions ended at this thing. John Collison from Stripe is there and Sundar, CEO of Google is there. And John, sorry, Patrick goes up to Sundar and they can talk about anything. Stripe wasn't the behemoth, it was now at that point, but it was still a significant company, was up and coming. And what does Patrick want to talk to the Sundar about? It's in the Gmail app, the dragging of people. When you reply all to a message, you often want to change the two recipient to CC and move someone from CC to two or something like that. And just how physically the degree of dexterity that's required to do that inside of the Gmail app is very high. It still hasn't been fixed, but it really struck me that Patrick could have asked for anything. It could have been any talk, it could have been a partnership. It was so irritating to him that it worked like this, he couldn't quite get over it. So anyway, back to bashing on Google, who in many respects do an incredible job and there's all kinds of amazing stuff they do on blah, blah, blah, but the Gmail actions on an individual email are broken into two very long menu items that are different. And one of them doesn't exist on either menu. There is an unlabeled icon is the only way to do it, and that's to mark something as unread once it's read. I have no idea why some of the actions are in one menu and some of the actions are in another menu. I think it's because some of them have to do with an individual email and some of them have to do with the whole thread, but it doesn't seem very consistent. Every possible thing is listed there in one place. And so it becomes incredibly difficult to use because sometimes you have to tap in both menus, read all of the options, and say, "Okay, I've used the

process of elimination and it's not here, so it must be there." Uber doesn't work like this anymore, but when I first brought this up to people inside of Slack, there was a moment when the Uber app, when you opened it was just, "Where would you like to go?" And other. And other was everything like change your payment method, set your location, anything you could do in Uber. And that was perfect because almost all the time people just wanted to choose where they wanted to go. Sometimes you wanted to change where your pickup was because you weren't there yet or whatever. And that was just like, what could be simpler than, "I'm going to tell you where I want to go or I'm going to achieve something else."? I really tried to push people to what is the thing that people, or what is the two things or what is maybe three things that people could want to do here and then put everything behind other. And then if it takes them eight clicks or taps to do something, but every single one is trivially easy, that's great. If you reduce that to two clicks or taps, but every part of it is this fraught decision where I'm opening all of the menus and trying to figure out which thing is the right thing, and the more, comparing three things to each other is this difficult four things, it's kind of geometrically more expensive to compare 15 different options all to the other to see if this is the one that you might want. That just becomes impossibly expensive. So to me, those are all really connected. And if people could get over the idea of reducing friction as the [inaudible 00:43:42] or reducing the number of clicks or taps to do something and instead focus on how can I make this simple? How do I prevent people from having to think in order to use my software? How can I make this trivially easy? One last example, because this was really influential for me. So I was going back and forth in Vancouver in San Francisco at the time when we were talking about all this inside of Slack, and I was behind a teenager in line aboard the plane and it was like, we're on the jet way. It took a long time. And I was watching her use Snapchat and it was insane. She was tapping at least four times a second, sometimes six or seven times a second. It was like dismissing stories and doing stuff. But there was a fluidity to it because everything was like, do I want to see this again? Do I want to see the next story from this person? Do I want to switch to a different person? Instead, a notification came up, she answered someone's thing, she took a selfie of herself and everything was just like... So she was tapping four times a second for six minutes. I mean, probably there was some breaks in there. And that was the highest and best use of Snapchat for a 15 year old girl in 2016 or whenever that was. And imagine if the goal was to try to make her tap less, how much of an impediment it would've been to the experience that both her and Snapchat wanted to create?

中文翻译:

大概八年前，在一次会议后的酒吧里，Stripe 的 John Collison 和 Google 的 CEO Sundar 都在。Patrick (Stripe 创始人) 走向 Sundar，他本可以聊任何事，比如合作。但他聊的是 Gmail App 里拖拽联系人的操作。当你“回复所有人”时，常想把收件人改成抄送，或者反过来。在 Gmail 里完成这个操作需要的指尖灵活性极高。这事至今没改，但让我震惊的是，Patrick 竟然如此在意这个细节，以至于要在这种场合提出来。回到吐槽 Google，Gmail 的操作被分成了两个很长的菜单，而且逻辑不通。有些操作甚至不在菜单里，只有一个没标签的图标（比如设为未读）。我完全不明白为什么有些操作在这个菜单，有些在那个菜单。所有可能的选项都堆在一起，导致你得翻遍两个菜单，用排除法找功能。以前的 Uber App 做得很好：打开后只有“你想去哪？”和“其他”。“其他”里包含改支付方式、设位置等。这很完美，因为绝大多数时候人们只想打车。我一直推崇：找出用户最想做的 1 到 3 件事，剩下的全藏进“其他”。哪怕用户需要点 8 次才能完成某个操作，但如果每一步都极其简单、不用思考，那也比点 2 次但每一步都要在 15 个选项里纠结要好得多。比较 15 个选项的认知成本是几何级增长的。最后举个对我影响很大的例子：在温哥华飞旧金山的登机口，我排在一个青少年后面，看她玩 Snapchat。太疯狂了，她每秒钟至少点 4 次，有时甚至 6、7 次。她在刷故事、回复消息、自拍，动作极其流畅。她连续这样点了 6 分钟。那是 2016 年一个 15 岁女孩使用 Snapchat 的最高境界。想象一下，如果 Snapchat 的目标是“减少点击次数”，那会多么严重地阻碍她和产品想要创造的那种流畅体验？

[00:45:06] Lenny Rachitsky

English:

It's so fun to listen to this and the examples you gave of, it gives us a lot of insight into the way your mind works of just constantly unsatisfied with the way other products work with your product. And I think that's core. Patrick is a good example of Stripe. I feel like that's a recurring theme with very successful product leaders is just constantly unsatisfied and unhappy with how things work.

中文翻译:

听你讲这些例子太有意思了，这让我们深入了解了你的思维方式——对其他产品以及自家产品的运作方式永远感到不满。我认为这是核心。Stripe 的 Patrick 也是个好例子。我觉得这是非常成功的产品领导者的共同主题：对现状永远感到不满和不悦。

[00:45:27] Stewart Butterfield

English:

Yeah.

中文翻译:

是的。

[00:45:28] Lenny Rachitsky

English:

I love just even the way you summarize this, just a really good reframing of, instead of obsessing with reducing friction and reducing steps, instead think, how do I reduce the amount of thinking the user has to do? I've never heard of it described as, you have to think about the ATP and glucose being used to actually think, and your goal is to reduce that versus let's just reduce friction, reduce clicks.

中文翻译:

我非常喜欢你总结的方式：不要痴迷于减少摩擦和步骤，而是思考“我如何减少用户必须进行的思考量？”我从未听过有人从消耗 ATP 和葡萄糖的角度来描述思考，并把减少这种消耗作为目标，而不是单纯地减少点击。

[00:45:52] Stewart Butterfield

English:

Yeah. I think in my more cynical examples, I would say to people, "Stop what you're doing for a second, close your eyes, take a couple of deep breaths, and then pretend that you're an actual human being. And open their eyes again, and then look at this thing and see, can you figure out what it's supposed to do or say. Or what action you're supposed to take or what the impact will be if you take that action. There's a whole nother related cycle. But before I get into it, I know that I am verbose. I want to wrap up your last example of people being unsatisfied. So here's the quote that I was trying to find. This is 2014, so like that was the year that Slack actually launched officially in February. And this is now near the end of the year. I was interviewed by MIT Technology Review and asked if we were working to improve Slack. I said, "Oh God, yeah. I try to instill this into the rest of the team, but certainly I feel like what we have right now is just a giant piece of shit. It's just terrible and we should be humiliated that we offer this to the public. Not everyone finds that motivational though." So I came into the office the next day and people had printed out on like 40 pieces of 8.5 by 11 paper that quote, and pasted it up on the wall. But to me that was like, you should be embarrassed by it. It should be a perpetual desire to improve. You should probably be like, "Oh, this is great," and you could be proud of individual pieces of work. But in the aggregate, if you can't

see almost limitless opportunities to improve, then you shouldn't be designing the product, or you shouldn't be in charge of the company, or you shouldn't almost nothing. Again, you could reduce it down to a tiny feature is anywhere close to perfect. And if A, that's acknowledged freely inside the organization. And B, people think about continually improving as the goal. And that could be like Six Sigma Toyota, Kaizen, that kind of side of thing. Or it could be that story that... I can't remember his name right now. The guy who started Bridgewater tells about Michael Jordan-

中文翻译:

是的。在我那些更愤世嫉俗的例子里，我会对员工说：“停下你手头的事，闭上眼，深呼吸，然后假装你是一个正常的人类。睁开眼，看看这玩意，你能搞清楚它是干嘛的吗？你能看懂它在说什么吗？你知道该点哪吗？你知道点了之后会发生什么吗？”这又涉及另一个循环。在我展开讲之前（我知道我话很多），我想回应你刚才说的“不满足”。我找到了那段话：那是 2014 年底，Slack 刚在 2 月发布。我接受采访被问到是否在改进 Slack，我说：“噢天哪，是的。我试图把这种想法灌输给团队，但我确实觉得我们现在的產品就是一坨巨大的狗屎。它太烂了，把它拿给公众用简直是我们的耻辱。不过，并不是每个人都觉得这话很励志。”第二天我回办公室，发现大家把这段话印在 40 多张纸上贴满了墙。但对我来说，你应该感到羞愧，这应该是一种永恒的改进欲望。你可以为某个具体的工作感到自豪，但总体而言，如果你看不到近乎无限的改进空间，你就不该设计产品，不该管理公司。没有任何一个小功能是接近完美的。如果组织内部能坦诚承认这一点，并把持续改进作为目标（就像丰田的精益生产 Kaizen），那就对了。或者像桥水基金创始人瑞·达利欧在书里讲的迈克尔·乔丹的故事——

[00:48:11] Lenny Rachitsky

English:

Ray Dalio.

中文翻译:

瑞·达利欧 (Ray Dalio)。

[00:48:12] Stewart Butterfield

English:

Yeah, Ray Dalio in his book talks about Michael Jordan learning to ski. Every time he messed up, he wanted the ski instructor to tell him exactly what he was doing wrong. Because to him, every one of those was a gem that he could collect, and he could actually become a good skier. And what he wanted to do was become a good skier. That requires a lot of trust inside the organization. But if you can get to the point where like, "Hey, we are trying to find improvements. We're trying to be critical because you're trying to make this as great as it can possibly be." And not always, not with every person, but most of the time with most people, you can get them to the point where that really direct criticism is actually motivational. It is like people are grateful to have the feedback, whether that's coming from their peers inside the company or from end users of the product. Because you realize, oh yeah, that is bad and we should fix it.

中文翻译:

对，瑞·达利欧在他的书里提到迈克尔·乔丹学滑雪。每次他摔倒或出错，他都希望教练准确告诉他错在哪。因为对他来说，每一个错误都是他可以收集的“宝石”，能让他真正成为滑雪高手。他想变强。这需要组织内部有极大的信任。如果你能达到这种状态：“嘿，我们是在寻找改进点，我们保持批判性是因为我们想让它尽可能完美。”虽然不是每个人都能接受，但大多数时候，这种直接的批评反而是励志的。人们会感激这些反馈，无论是来自同事还是用户，因为你意识到：“噢对，这确实很烂，我们得修好它。”

[00:49:10] Lenny Rachitsky (Sponsor: Lovable)

English:

This episode is brought to you by Lovable. Not only are they the fastest growing company in history, I use it regularly and I could not recommend it more highly. If you've ever had an idea for an app but didn't know where to start, Lovable is for you. Lovable lets you build working apps and websites by simply chatting with AI. Then you can customize it at automations and deploy it to live domain. It's perfect for marketers, spinning up tools, product managers prototyping new ideas, and founders launching their next business. Unlike NoCo Tools, Lovable isn't about static pages. It builds full apps with real functionality, and it's fast. What used to take weeks, months, or years, you can now do over a weekend. So if you've been sitting on an idea, now is the time to bring it to life. Get started for free at [Lovable.dev](https://lovable.dev). That's lovable.dev. This makes me think about, let's call it a rant that you have about how it takes a lot of work to make anything work at all. That just the default state is not working. Can you just share what you share there.

中文翻译:

本期节目由 Lovable 赞助。他们不仅是历史上增长最快的公司之一，我也经常使用它，强烈推荐。如果你有过开发 App 的想法却不知从何下手，Lovable 就是为你准备的。Lovable 让你只需通过与 AI 聊天就能构建可运行的 App 和网站。你可以自定义自动化流程并将其部署到线上域名。它非常适合需要快速制作工具的营销人员、需要原型设计的 PM 以及想要启动下一个业务的创始人。与无代码工具不同，Lovable 不仅仅是静态页面，它能构建具有真实功能的完整 App，而且速度极快。以前需要数周或数月的工作，现在一个周末就能完成。如果你有一个想法，现在就是实现它的时刻。在 [Lovable.dev](https://lovable.dev) 免费开始。这让我想起你的一段“吐槽”：让任何事情运转起来都需要付出巨大的努力，而事物的默认状态其实是“不运转”。能分享一下你的观点吗？

[00:50:13] Stewart Butterfield

English:

Yeah. I mean, so this is a lot to do with, and maybe this is more recently, it shows up in politics a lot for me. But by the way, if anyone listening to this can help me find this tweet store from somewhere between 2016 and 2020, I don't have a precise idea. And it was this guy's thread about how hard it was to get a stop sign set up. And I believe it was in response to someone claiming that Bitcoin is going to replace US dollars, something about crypto. And his point was like, here's what happened when we tried to get a stop sign put up on a residential street in my neighborhood. And the literal years it took, and the number of agencies that were involved. Like the engineering department, traffic planners, the HOA, and... I don't remember all of the organizations because, and I did that I could search better and find this again. Because it was truly a masterpiece of how difficult it is to get a stop sign put up in most places. The message that I hear from most politicians, and unfortunately this works really well, is things should be good. But they're not because someone is doing something bad, which is preventing the goodness. So billionaires are making things unaffordable. Or immigrants are taking your jobs. Or lazy freeloaders are sucking off a government teat, and causing us all have to pay more taxes, or something like that. The reality is almost nothing works. It's actually another call. I said in this case, John has a great encapsulation of this and I'm sure you're familiar with it, like that. It ends with the world as a museum of passion projects. Because for anything to get done at all requires not just the resources and effort required to instantiate that thing in the real world, but all of the politicking and the sociology and the convincing. And there's a book called *Why Nothing Works Recently*, which is like, it's not an... I'm sorry to the author, if they... I doubt they're listening, but just it's not like an amazingly written book. I found it a little bit repetitive, but the content was really incredible, just explaining why it's so hard. And how there's

this progressive increase in the number of vetoes that are available for any kind of course of action and how difficult it is... And this shows up in permitting for new construction and stuff like that. But also shows up obviously inside of organizations. And the challenge is that people, A, I think this is evolutionary biological. It's hard for us to understand the world, except by anthropomorphizing it. And so if it didn't rain this year, it's because a God is mad, and probably because we didn't sacrifice enough goats or something last year. It's hard for people to understand just that, wow, weather is incredibly complex and chaotic, and ecosystems and climatology, and all that. Same thing with the world. Like if I am struggling to pay all of my bills and be able to afford a little bit of luxury in the sense of location or a present for my kids or whatever, it's got to be somebody's fault. There has to be a decision that's made somewhere. And the reality is everything is so complicated. Everything is so multivariate, it's not satisfying. It's a terrible political message. It's much easier to say that there is like, oh, we understand why things are bad in the way that you're concerned about. And it's turns out that it's some someone's decision, and because of them it's bad. And so if we got rid of them or were able to overcome their decision, overturn it, and institute our own thing, then things would be good for you. And this really to me shows up inside of those organizations as well. I'll pause there.

中文翻译:

是的。这与我最近对政治的观察有关。顺便说一下，如果有人能帮我找到 2016 到 2020 年间的一条推特，那是一个关于“设立一个停车标志（Stop Sign）有多难”的帖子。它好像是在反驳“比特币将取代美元”之类的言论。作者描述了在他家附近的住宅区设立一个停车标志经历了多少年，涉及了多少部门：工程部、交通规划部、业主协会（HOA）等等。那简直是描述“办事难”的杰作。大多数政客传达的信息是：世界本该是美好的，但现在不好是因为有坏人在捣乱。比如亿万富翁让物价飞涨，或者移民抢了工作，或者懒汉在吸政府的血。但现实是：几乎没有任何事情是能顺畅运作的。Stripe 的 John Collison 有个精辟的总结：“世界是激情项目的博物馆。”因为任何事情要办成，不仅需要资源和努力，还需要大量的政治博弈、社会学博弈和说服工作。最近有一本书叫《为什么一切都行不通》（Why Nothing Works），虽然写得有点重复，但内容很棒，解释了为什么现在任何行动面临的“否决权”都在不断增加。这体现在建筑许可上，也体现在组织内部。人类有一种进化本能，喜欢把世界“人格化”来理解：如果今年没下雨，那一定是神生气了，因为我们去年没祭祀够。人们很难理解天气是极其复杂且混沌的系统。同样，如果我付不起账单，那一定是某个人的错，一定是某个地方做错了决定。现实是，一切都太复杂、变量太多，这个答案并不讨喜，也不是一个好的政治口号。说“这是某人的错，干掉他世界就好了”要容易得多。这种心态在组织内部也随处可见。

[00:54:25] Lenny Rachitsky

English:

I know kind of along those lines, you're a big believer in something called Parkinson's Law.

中文翻译:

我知道沿着这个思路，你非常相信所谓的“帕金森定律”（Parkinson's Law）。

[00:54:31] Stewart Butterfield

English:

Yeah. So the original of that is, I think it's 1956. It's an article in The Economist by Parkinson. And the Maxim is work expands to fill the time available for its completion. And the way that it shows up, this is a little bit subtle. So like one of the things I found, since I don't have a job is there's much less time pressure. And that maxim, like if you want something done, give it to a busy person. The inverse is also true that like, if you're not that busy, wow, basic things take a really long time. And so Parkinson actually starts out with his example of writing and posting a letter. And I don't remember who he used with the

first example, but someone who's incredibly busy and has all these things they have to respond to. And then another case like a retired robot who has all the time in the world. It takes her a long time to write the letter. It takes her a long time to put it in the envelope, and then you go to the post office and post it. But the real meat of it is, for me later when he talks about the size of the organization, and he uses a bunch of examples. This is again 1950s, and he's British, so he's looking at the Royal Navy. And specifically he's looking at a chart that shows the relationship between the number of capital ships in the Navy, the number of sailors, and the number of administrators. And very familiar graph for people looking at any part of government. Any part of the relationship between the number of administrators at a university and the number of students and faculty, teaching faculty. Where it's like, okay, the number of ships goes like this and the number of sailors is looking right along with it. And the number of administrators goes like this. And the reason this ties into the work expands to fill the time available for its completion is people hire, and they train. And here's the sad truth for anyone running a company is there are exceptions. There's certain types of engineers that are an exception to this. But the overwhelming majority of people you hire want to hire more people who report to them. And it's not because they're evil, and it's not because they're stupid. In fact, they're smart because everyone knows that the number of people who report to you correlates with your career trajectory, the amount of money that you're paid. The amount of authority you have inside the organization and on and on and on. So we would hire 27 Royal product managers in Slack who immediately want to hire someone. It's like, what the hell? What would that person do? And they articulate it this way, but essentially it's like, "Well, that person would do the product management and then I would do strategy."

中文翻译:

是的。这最早源于 1956 年《经济学人》的一篇文章。核心格言是：“工作会自动膨胀，占满所有可用的完成时间。”这表现得很微妙。比如我现在没工作，时间压力小了，我发现：如果你想把一件事办成，就把它交给一个忙碌的人。反之，如果你不忙，基本的小事也会花掉很长时间。帕金森举了写信寄信的例子：一个大忙人很快就能写完寄走；而一个有大把时间的退休老人，写信、装信封、去邮局能磨蹭一整天。但对我来说，最核心的部分是他讨论组织规模。他观察了 50 年代的英国皇家海军：军舰数量在减少，水兵数量也在减少，但行政管理人员的数量却在飞速增长。这在政府、大学（管理人员与师生的比例）中随处可见。为什么？因为人们雇人、培训。对于经营公司的人来说，一个悲哀的现实是：除了极少数工程师外，绝大多数你雇佣的人都想雇佣更多向他们汇报的人。这不是因为他们坏或蠢，相反，他们很聪明，因为大家都知道：下属人数与你的职业轨迹、薪水、权力和地位直接挂钩。所以我们在 Slack 招了 27 个产品经理，他们立刻就想再招人。你会问：“那个人干嘛？”他们会说：“噢，那个人负责具体的产品管理，我负责战略。”

[00:57:11] Lenny Rachitsky

English:

Classic.

中文翻译:

太经典了。

[00:57:12] Stewart Butterfield

English:

It's really, I think the essential thing to understand about this is it's not because people are evil, and it's not because they're stupid. And it's to me, very related to everything is complex. And if maybe this is my butterfly's law, I haven't thought about this way before. But I tweeted this a very, very long time ago like if you... Everything is simple if you have no idea what you're talking about. So the other side of that is like if

something seems simple, probably you don't understand it. And there's obvious exceptions to that. But for anything that involves a large organization or a lot of human beings, if the problem seems simple, you don't get it. So every budget process, no head of engineering know, head of sales, no CFO, no GC, who's ever going to come back and say, "Oh, I've actually think next year we can just hire fewer people. Or we're going to keep it flat or we're going to shrink through attrition because we don't need any more people to do what we're doing." Not because they're evil, not because they're stupid, but it's almost overpowering impulse inside the organization that often leads to disastrous results. And so there's an... I'll give one example from Slack's history, and I have tried in the past to disguise this example so that no one feels bad about it but I... Unfortunately, the specifics are so important to the example that it's not disguised and so I'll just reiterate that the people involved aren't stupid or evil. And one example that's from the outside. So the example inside of Slack was we introduced threads, which was the ability to reply to a message inside of a channel. And let's say you, Lenny, post a message. I, Stewart reply to it. You will automatically get a notification. And now Sarah later on replies to the same message. Both you and I, as people who have pushed in that thread will receive a notification that there's been more activity, and so on. So like every single time anyone replies to it. So when the feature first was released or like when we did the final product review before it was released, the input box was pre-populated with at the person before you in the thread. And I was using the feature and I would put the insertion point there, select all delete, and then start writing my message. And even if I wanted to add someone specifically, I almost never wanted to start my sentence with at that, because it just made it hard to reference what they were saying before. So I said, "Get rid of this because, A, I think most people won't use it. Or if they did want to add someone, they're not going to want to do it at the beginning of the sentence. And by the way, you're teaching them to use the product wrong. Because it's important that everyone understand that every previous poster in this thread will automatically receive a notification unless they've figured it." So okay, we release it. Six months goes by and suddenly the at thing comes back. And so I messaged someone around the team and I said, "Hey, there's been a regression. This is super weird. I don't know what happened. But the at thing came back." And they said, "Oh no, this is on purpose. We did a bunch of research." And so I was like, "What?" And I went through this and it was, if I recall correctly, it wasn't even P-95 certainty on this analysis. But it was something like when we do this, threads are 2.17 messages long, versus 2.14 messages long on average for when we don't do it. And so first of all, why is a longer thread better? Like maybe a shorter thread is better? It can be fewer messages that people have to go back and forth. Also, that's such a tiny difference. Also, again, I don't remember the actual statistical analysis, so I'm not going to claim that it was incorrect. But I'm pretty sure this was outside the bounds of certainty that they can have. But the real thing was, oh my God, so you guys put flags into the product, you A-B tested it. You did the instrumentation. You created tables in the database or whatever we're using to record all of that. You wrote queries to pull that. You created charts based on that data. You had meetings to discuss it. And just kind unpacking all of the things that would've had to happen for this to come back. And it's like thousands of person hours at a minimum, because any feature change at that scale of organization, it's involving like a dozen people. Engineering, QA, analytics teams, project managers, user research and stuff like that. The problem with that, so I think it was a bad idea, right? But the problem with that was the difference that you could possibly achieve between having this feature and not having this feature is like this much whatever units you want. The cost of doing the analysis was this much. So it's guaranteed to be a loser. Like there's just, there's no world in which anyone could imagine putting the at previous respondent in the thread at the beginning of the message could possibly make that much of a difference to the quality of Slack, and how much utility it provides for people and all of that. But you know that to put the feature flags in, to ship new versions of the product, to put the instrumentation in. To have it all the API calls to record every action that people take to do all the analytics, to create the dashboard. To put paste a screenshot of that into a Google Slides presentation. To send the invitations to the meeting, to reschedule the meeting because someone couldn't make it. To have everyone sit down and look at the thing. Like guaranteed loser. And I know that Fareed told you to ask me about this hyper realistic work-

like activities. And so here's my grand theory. Hyper realistic work-like activities goes along with this other concept called known valuable work to do. And when I say known, I mean both you know what it is and you know that it's valuable. And the problem with almost every organization at the very beginning, you have an enormous amount of work that you know what to do, and you know that it's going to be valuable. So like starting a business, open a bank account. Because there's almost infinite general value of opening a bank account. You have to do it. It's very simple to do. And so at the very beginning of any startup, they're like, "I'm creating a user's table, and I'm doing sorting passwords," and you're doing all the things that are kind of absolutely necessary. And everyone knows exactly what they are. And so everyone's going to work in the morning and they're like right on. And I have 10 things to do, and every single one of them is something I know how to do. And it's definitely going to be valuable. Time goes on. And the relationship between the supply of work to do and the demand for doing work just starts to change. More and more people get hired. Every product manager wants to hire a junior product manager. Every new person, the first person you bring in on the risk and compliance team is like, "Oh my God, there's so many risks and things we have to be compliant with. We better hire more people on my team to do more risk and compliance work." Which probably to some degree is right. But we're going to have more and more of those people and they're going to call meetings with each other. And now suddenly you have all these people with work to do and you've done all the easy obvious stuff. And now your questions are like, "God, should we do FedRAMP high and make a Slack version? Which is going to require us to have wholly separate physical infrastructure for the hardware that runs the software? And also a whole different operations team, which has only US citizens on it? What is the possible number of dollars that we could make from doing this? And how much complexity is going to be when we want to do updates to the software because we update two totally separate independent systems and rec." It just gets out of whack, and so people end up... Like if you hire 17 product marketers, you're going to have 17 product marketers worth of demand for work to do. And if you don't have sufficient supply of product marketing work to do, they're just going to do other stuff. Again, very important, not because they're stupid, not because they're evil. But because they're like, I'm a product marketer and I want to be recognized for my work. And my spouse has criticized me because they take, I should have already got promoted in the last cycle, and they really got to demonstrate some wins here and whatever it is. And so people are like calling meetings with their colleagues to preview the deck that they're going to show in the big meeting to get feedback on whether they should improve some of the slides. And that hyper-realistic work-like activity is superficially identical to work. Like we are sitting in a conference room and there's something being projected up there, and we're all talking about it. And that's exactly what work is. Hopefully not all of work for everyone inside of your company. But that's exactly what we do when we're working. But this is actually a fake bit of work, and it's so subtle that I'll do it. Our board members will do it. Every executive will do it. And the further you are from having all of the context and all of the information and the decision-making authority and stuff like that, the easier it is to get trapped in this stuff. And people will just perform enormous amounts of hyper-realistic work-like activities, and have no idea that that's what they're doing. And the result of that, I guess, is that if you are a leader, if you're manager, director, an executive, you're the CEO, it's on you to ensure that there is sufficient supply of known valuable work to do. And there almost always is, but it's creating the clarity around that. Creating the alignment. Making sure everyone understands it, but that's what they're supposed to be doing, and then obviously doing it.

中文翻译:

理解这一点的关键在于：这不是因为人们坏或蠢。这与“一切皆复杂”有关。我很久以前发过一条推特：“如果你完全不知道自己在说什么，那么一切都很简单。”反过来说，如果某件事看起来很简单，那你可能还没理解它。在大型组织中，如果问题看起来很简单，那你肯定没搞懂。在预算流程中，没有任何一个部门主管会说：“明年我们可以少招点人，或者通过自然减员缩小规模。”组织内部有一种强大的冲动，往往导致灾难性的后果。举个 Slack 的例子（我不针对任何人）：我们推出了“会话串”（Threads）。当你回复某人时，系统会自动

@对方。我当时觉得这很烦，因为我回复时几乎从不想在句首加个 @，这让引用变得很困难。而且系统本来就会自动通知参与者，没必要多此一举。于是我要求去掉这个功能。半年后，这个 @ 功能又回来了。团队告诉我：“我们做了大量研究。”结果显示：加上 @ 后，会话串平均长度从 2.14 条变成了 2.17 条。首先，更长的会话串真的更好吗？其次，这个差距微乎其微，甚至在统计学上都不一定显著。但真正让我震惊的是：为了让这个功能回来，你们埋点、做 AB 测试、建数据库表、写查询、画图表、开会讨论……这至少耗费了数千人时。在那种规模的组织里，这涉及十几个人的协作。这种分析的成本远超它能带来的价值。这就是我说的“超写实类工作活动”（Hyperrealistic work-like activities）。它与“已知的有价值工作”相对。在创业初期，所有工作都是已知的且有价值的（比如建用户表、开银行账户）。但随着人越来越多，显而易见的工作做完了，大家就开始找事做。如果你招了 17 个产品营销，他们就会产生 17 个人的工作需求。如果没有足够的实际工作，他们就会开始开会、预演 PPT、讨论幻灯片细节。这种活动在表面上看起来和“工作”一模一样：大家坐在会议室里，看着投影，讨论问题。但它其实是“假工作”。这种现象非常微妙，连我、董事会成员和高管都会掉进去。作为领导者，你的责任是确保有足够的“已知的有价值工作”供应，并建立清晰的优先级和共识。

[01:07:26] Lenny Rachitsky

English:

Amazing. I could listen to Stewart's rants all day. Hyper-realistic work-life activities. We need to coin this-

中文翻译:

太精彩了。我可以听 Stewart 吐槽一整天。“超写实类工作活动”，我们需要把这个词推广开来。

[01:07:34] Stewart Butterfield

English:

Unfortunately, it doesn't make a good acronym. It's pretty ugly.

中文翻译:

可惜它缩写不好听，挺丑的。

[01:07:36] Lenny Rachitsky

English:

Okay. [inaudible 01:07:37] it a try. And just to close the loop on that, the solution is the leader recognizing this is happening and stopping it. Telling people why are we spending time on this thing that is not going to get us anywhere?

中文翻译:

好吧。总结一下，解决方案是领导者要意识到这种现象正在发生并阻止它。告诉大家：为什么我们要把时间花在这些毫无进展的事情上？

[01:07:48] Stewart Butterfield

English:

Yeah. And what you just said probably isn't the best way because that sounds like you're chiding them, and they're dumb. When it's actually your responsibility to make sure that there's sufficient clarity around what the priorities are, and explicitly saying no to things upfront and stuff like that. Rather than

merging and say like, "Hey, you guys are a bunch of idiots wasting your time on this thing that doesn't matter." Whose fault is it? It's the manager's fault. It's the VP of whatever's fault. It's the CX, whatever, it's the C... Ultimately, it's the leader of the organization that has the responsibility to make sure that there is sufficient known-valuable work to do. And that's actually harder than it might appear.

中文翻译:

是的。不过你刚才说的方式可能不是最好的，因为那听起来像是在责备他们蠢。实际上，确保优先级足够清晰、提前明确拒绝某些事情，这是领导者的责任。而不是事后冲进去说：“嘿，你们这群白痴在浪费时间做没意义的事。”这是谁的错？是经理的错，是副总裁的错，是高管的错。最终，组织领导者有责任确保有足够的“已知的有价值工作”可做。这比看起来要难得多。

[01:08:32] Lenny Rachitsky

English:

Okay. Before we run out of time, I want to touch on two other topics. One is, when people think of Stewart Butterfield, I think a lot of people think of, We Don't Sell Saddles Here. Your legendary Medium post that is just, I don't know, it's become a historic piece of literature in the annals of product building and in startups. I haven't heard people ask you much about this recently. So let me just ask a couple of questions. What was the reason you put that out? What was the backstory on writing that memo? Why was it necessary?

中文翻译:

好。在时间结束前，我想聊另外两个话题。第一，当人们想到 Stewart Butterfield 时，很多人会想到那篇《我们这里不卖马鞍》(We Don't Sell Saddles Here)。那是你发在 Medium 上的传奇文章，已经成为产品构建和创业史上的经典文献。最近没怎么听人问过你这个，所以我想问：你当时为什么要发那篇文章？写那篇备忘录的背景是什么？为什么它是必要的？

[01:09:01] Stewart Butterfield

English:

Well, it really was an internal memo and there's a bit of a digression. One of the crappy things about Slack is if all your corporate communication is on email, depending on exactly how it works and what system you use, you probably walk away with an archive of everything you said at Company X. If it's Slack, once you're turned off, you lose access to all that history. And so it's kind of like, "Oh, man. If I had only exported all of my messages before I left, I would have all this stuff," but that was absolutely verbatim. I did not change a word of what I said inside the company. Well, I think we were still eight people. Maybe at most 10, but I think it was eight people.

中文翻译:

那确实是一篇内部备忘录。顺便说句题外话，Slack 有个糟糕的地方：如果公司沟通全在邮件里，离职时你可能还能带走存档；但在 Slack 里，一旦账号被关，你就失去了所有历史记录。所以很多人离职前会想：“噢天哪，我要是导出了所有消息就好了。”那篇文章是绝对的原话，我一个字都没改。当时公司大概只有 8 个人，最多 10 个。

[01:09:45] Lenny Rachitsky

English:

It was before Slack launched even.

中文翻译:

那甚至是在 Slack 正式发布之前。

[01:09:47] Stewart Butterfield

English:

Yeah, it was before Slack launched. It was when we're doing private beta. And the point of it was to start to instill those ideas as early as possible and really create this alignment inside of that small team so that it could persist to survive as we grew and scaled. Yeah, that was the idea.

中文翻译:

是的，在发布之前，还在内测阶段。目的是尽可能早地灌输这些理念，在这个小团队内部建立共识，以便在公司成长和扩张的过程中，这些理念能够延续下去。

[01:10:11] Lenny Rachitsky

English:

And the gist, just for people that aren't super familiar with it, but we'll link to it, is just it's not enough just to build a great product. You just as much have to put effort into communicating what this does for them, the problem this is solving for them, the outcome this is going to achieve for them. Is that a good way to think about it?

中文翻译:

对于不太熟悉这篇文章的人（我们会附上链接），其要点是：仅仅打造一个伟大的产品是不够的。你必须投入同样的精力去沟通：这个产品能为用户做什么？它解决了什么问题？它能带来什么结果？这样理解对吗？

[01:10:28] Stewart Butterfield

English:

Yeah. And again, comparing it to beer or cars, beer goes back to pre-civilization. Cars were obviously [inaudible 01:10:38], but at some point you had to convince people why they would want a car instead of a horse. For your new AI-based recruiting tool or your calendar app or whatever, there's some reason why you think that people should use yours instead of the thing that they're using now, which might be a wholesale one-for-one replacement, or more often is a change in the way that you're working that has a bunch of other adjacencies and you want to expand into these other categories. You're not just responsible for creating the product, but also, to a certain degree, creating the market. There's this book, *Positioning*, which is an absolute classic. It's very short. I would recommend everyone read it, where the point of it is, from my perspective, it's almost impossible to create a new idea in someone's head. It's much easier to take a couple of existing ideas and put them together. So it's much easier to say it's like *Jaws* meets *Star Wars*, or it's *Uber* for *Pets* or something like that, than to come up with an actual new idea. But you have to do that because if your thing is different in any significant way from the alternatives, you're not just creating the product. You're creating the market. They're really kind of one and the same.

中文翻译:

是的。再拿啤酒或汽车做比喻：啤酒的历史可以追溯到史前文明；而汽车出现时，你必须说服人们为什么想要汽车而不是马。对于你的 AI 招聘工具或日历 App，你必须给出一个理由，为什么人们要放弃现有的工具改用你的。这可能是一个直接的替代，但更多时候是工作方式的改变。你不仅负责创造产品，在某种程度上，你还在创造市场。有一本经典的书叫《定位》(Positioning)，非常短，我建议每个人都读读。它的核心观点是：在别人脑子里凭空创造一个新概念几乎是不可能的，把两个已有的概念组合起来要容易得多。比如“《大白鲨》遇上《星球大战》”或者“宠物界的 Uber”。如果你的产品与替代品有显著不同，你就是在创造市场。这两者其实是一回事。

[01:11:56] Lenny Rachitsky

English:

The reason I wanted to touch on it is I think still people continue to not listen to this advice and continue to over-invest in more features, more products, things like that. Just the specific example of, "We don't sell saddles here," just to quickly communicate this to folks, and correct me if I'm missing anything, is just instead of, "Hey, look at this amazing saddle we've bought," which you want to communicate as, "Here, go horseback riding. Look at this incredible experience you can have." And then they decide, "Oh, shit. I need to go buy a saddle to do that."

中文翻译:

我之所以想提这个，是因为我觉得人们依然不听这个建议，继续过度投资于更多功能和产品。关于“我们这里不卖马鞍”的具体例子，我的理解是：不要说“嘿，看我们做的马鞍多棒”，而是要说“去骑马吧，看这种体验多美妙”。然后用户会意识到：“噢，该死，我得买个马鞍才能去骑马。”

[01:12:23] Stewart Butterfield

English:

Yeah. And 100%, that aspect of it is not original because I think that's something that marketers have done for a long time, certainly in the marcom and advertising. If you want to sell Harley-Davidson's, there are people who are going to geek out on the engines and stuff like that and the quality of the leather and stuff like that. But when you're selling the motorcycle, you're selling the open road and freedom and the wind in your hair. And if you're Lululemon, you are obviously selling yoga pants, but you're also selling health and aspiration and being the best version of yourself and a bunch of other stuff. Oh my God, I forgot the classic version of it.

中文翻译:

是的。100% 正确。这个观点其实并不原创，营销人员已经用了很久了。如果你卖哈雷摩托，虽然有人会钻研引擎和皮革质量，但你卖的是“开阔的公路、自由和拂过发梢的风”。如果你是 Lululemon，你卖的是瑜伽裤，但你同时也在卖“健康、向往和成为更好的自己”。天哪，我忘了那个最经典的的说法了。

[01:13:00] Lenny Rachitsky

English:

There's the ship ...

中文翻译:

那个关于造船的……

[01:13:00] Stewart Butterfield

English:

You're selling the screwdriver.

中文翻译:

你在卖螺丝刀（指卖孔而不是卖钻头）。

[01:13:04] Lenny Rachitsky

English:

Oh, yeah. The nail.

中文翻译:

噢对，钉子。

[01:13:05] Stewart Butterfield

English:

Yeah, the nail. Anyway.

中文翻译:

对，钉子。总之就是这个意思。

[01:13:07] Lenny Rachitsky

English:

Yeah, we missed that one. Well, there's the one I think about is instead of trying to convince men to build a ship, instill a yearning for the sea.

中文翻译:

是的，我们漏掉了那个。我想到的是：与其说服人们去造船，不如让他们向往大海。

[01:13:16] Stewart Butterfield

English:

Yes. Exactly. That's something that goes back in history.

中文翻译:

没错，正是如此。这是历史悠久的智慧。

[01:13:21] Lenny Rachitsky

English:

Okay. Let me ask you about pivoting. You are potentially the king of pivots. You started two companies both famously pivoted, both from video games, which is why I asked you about that at the beginning, into

very successful companies. I imagine many people come to you for advice on pivoting. Let me just ask when folks come to you asking, "Should I stick with my idea? Should I pivot?" what sort of advice do you find most helps them?

中文翻译:

好，问问关于转型（Pivot）的事。你简直是“转型之王”。你创办了两家公司，都经历了著名的转型，都是从视频游戏转型为极其成功的公司。我想很多人找你咨询转型。当人们问你“我该坚持我的想法，还是该转型？”时，你觉得什么样的建议对他们最有帮助？

[01:13:43] Stewart Butterfield

English:

Yeah, I mean, I think it's partly an intuition because obviously the decision is about, "Have you exhausted the possibilities?" and in the case where we were working on Glitch, this game where we used IRC for internal communication and we added a bunch of IRC which became the Proto Slack. I think Slack had an enormous advantage in the fact that we are working on this for several years without actually explicitly working on it and only doing the minimum number of features that were absolutely guaranteed to be successful in the sense that it was so irritating that we couldn't stand it anymore or such an obvious improvement that we couldn't help but take advantage of it. We still had \$ 9 million left and everyone still liked the game and we were all happy working on it, but I think by that point I had exhausted every non-verdacious long shot idea to make it commercially successful, and so I decided to abandon it. But the default advice for anyone in anything is persevere. It's like a kitten hanging off the branch and a poster says, "Hang in there." There's so many stories of, "So-and-so started out going door-to-door and was rejected by everyone and then suddenly there was Nike," or something like that and just, "If you stick with it long enough, you'll eventually be successful." I think you have to really be coldly rational. Some of this shows up in the book Thinking in Bets. Some of it's in Annie Duke's second book, the title of which I'm forgetting right now, but someone will know it.

中文翻译:

这部分靠直觉，因为核心在于：“你是否已经穷尽了所有的可能性？”在做 Glitch（那个游戏）时，我们用 IRC 进行内部沟通，并加了很多功能，那就是 Slack 的雏形。Slack 有个巨大的优势：我们在没有明确“做产品”的情况下已经用了它好几年，只添加那些因为“受不了了”而必须做的功能。当时我们还有 900 万美元，大家也喜欢那个游戏，但我意识到我已经穷尽了所有能让它在商业上成功的合理想法，所以我决定放弃。通常给人的建议是“坚持”，就像那张小猫挂在树枝上的海报写着“挺住”。有太多“某某人被拒绝了 100 次最后成功了”的故事。但我认为你必须保持冷酷的理智。Annie Duke 的书《对赌》（Thinking in Bets）里提到过这个。

[01:15:35] Lenny Rachitsky

English:

Yeah, Thinking in Bets, and then what was the second? I forgot.

中文翻译:

对，《对赌》，第二本叫什么来着？我也忘了。

[01:15:39] Stewart Butterfield

English:

She actually uses Glitch and Slack as an example of a smart fold basically. My expected value here has diminished to the point where this alternative looks more attractive. And the reason I say you have to be coldly rational about it is because it's fucking humiliating. I convinced so many and you have to convince so many people to get a company off the ground. You have to go to investors. You have to go to early employees and say, "You should leave your other job and come work for this because here's the incredible feature we're imagining." You have to go to the press and you have to make all these promises and you have users and you've committed things to the users and you've convinced them to give up their time for this thing. And so I think for a lot of people, it feels better to just keep doing it until it dies of suffocation due to lack of capital or something like that. Then just to admit, "Okay, I was wrong. This didn't work," and it's humiliating. It's painful. It's wrenching. It has a bad impact. When we shut down Glitch, there was a lot of people who loved it and would spend all of their free time and couldn't wait to get home from work to go play it more. And that was their community and the community just disappeared, all these people and all these identities that have been created. And obviously, people lost their jobs and people who had moved their families to a different city in order to take this job now weren't going to have a job anymore. So pivots aren't something I take lightly. I think it's very different to be like, "There's three us and we started making this app and then we pivoted to a different app." That doesn't even really count. If you're six months into something, you're still messing around. You're trying to figure out what it is that you're building. It's not really a pivot. Obviously in this case, it worked out great and there's survivorship bias and that doesn't mean that everyone should pivot all the time. But I think creating the distance so that you can make an intellectual, rational decision about it rather than an emotional decision is essential.

中文翻译:

她把 Glitch 和 Slack 作为“聪明止损”的例子。当当前路径的预期价值降低到一定程度，替代方案就显得更有吸引力。我之所以说要冷酷理智，是因为承认失败真的太他妈丢脸了。为了创办公司，你得说服投资人，说服早期员工辞职加入，对媒体许下诺言，对用户做出承诺。对很多人来说，一直耗到资金耗尽而“窒息死亡”，比主动承认“我错了，这行不通”要好受得多。承认失败是痛苦的、揪心的。当我们关掉 Glitch 时，很多热爱它的用户失去了社区，员工失去了工作，甚至有人为了这份工作刚举家搬迁。所以我并不轻视转型。如果你只有三个人，做了六个月就换方向，那不叫转型，那叫摸索。虽然 Slack 成功了，这里有幸存者偏差，但这不代表每个人都该随时转型。关键是拉开距离，做出理智的决策而非情感化的决策。

[01:17:50] Lenny Rachitsky

English:

I love, also, your piece of advice of just exhaust. Once you've exhausted all the ideas, that's a really good time to see what else is out there.

中文翻译:

我也很喜欢你关于“穷尽”的建议。一旦你穷尽了所有好主意，那就是看看还有什么其他机会的好时机。

[01:17:56] Stewart Butterfield

English:

Yeah, just all the good ideas.

中文翻译:

是的，所有“好”主意。

[01:18:00] Stewart Butterfield

English:

All the realistic. Yeah.

中文翻译:

所有现实的主意。

[01:18:03] Lenny Rachitsky

English:

Yeah. The point you made about just kind of persevering, I just had Melanie Perkins, CEO of Canva, in the podcast. 100 investors rejected her before somebody finally decided to invest and she just kept pushing.

中文翻译:

关于坚持，我刚采访了 Canva 的 CEO Melanie Perkins。在有人投资前，她被 100 个投资人拒绝了，但她一直在坚持。

[01:18:19] Stewart Butterfield

English:

Yeah. I think that's a slightly different example, right? She eventually believed in the concept of the product and in the vision. It was just trying to figure out the right articulation to get investors who ended up being obviously very, very happy.

中文翻译:

是的，我觉得那是稍微不同的例子。她始终相信产品理念和愿景，她只是在寻找正确的表达方式来说服投资人。显然，那些投资人最后非常开心。

[01:18:31] Lenny Rachitsky

English:

Extremely happy. Oh, geez. Okay. Maybe a final topic depending on how time goes. I want to talk about generosity. I talked to a bunch of people, as I said, that have worked with you and the number one theme that came up again and again and again when I asked them about you and what has stuck most with them is just generosity. So I'm going to read a few examples that I heard from folks that are examples of your generosity over the years. So one person shared that he needed a little money before Christmas and he said, "Stewart literally walked me out of the building, went to the cash machine, handed me \$500, told me to go home to my family." Other folks shared that, when you talked about Glitch just recently when you had to lay people off, you cried real tears when you were laying people off and then you spent an incredible amount of time helping them find new jobs and extending their severance pay and just taking it extremely, extremely seriously, much more than I think most people feel like CEOs do. Someone else shared that you paid 100% of employees health insurance to give them just fewer things to think about. When you went public, you basically created the best possible situation for employees, no lockup, direct listing. Also, with the structure of the Slack deal, people said that acquisition was very employee friendly. That's employees. There's also just the way you thought about customers. A few examples: You gave free credits to businesses who were struggling to pay the bills during COVID. You released this fair billing,

which I think was very innovative at the time, where you stopped charging people for seats they weren't using, even though they signed a deal to charge for those seats. A lot of times, you slipped release schedules because you just wanted to make features better and better for people. And I'll end with this quote: "Stewart is a leader who takes the responsibility he feels for his employees personally, and to which he extends the most generous circumstances he could muster. That feels worth celebrating." So first of all, I just want to celebrate you. I think it's really rare and inspiring to meet a leader like that. Clearly, you've had a lot of impact on a lot of people. I don't know exactly the question I want to ask, but I guess in what part is this intentional, just like, "This is how we win. I'm going to be very generous and help people because I know this will help long-term"? How much of this is just a [inaudible 01:20:48] and it's just the way you are as a person?

中文翻译:

非常开心。最后一个话题，我想聊聊“慷慨”（Generosity）。正如我所说，我采访了很多与你共事过的人，当我问起你最让他们印象深刻的一点时，答案反复出现：慷慨。我读几个例子：有人说他在圣诞节前缺钱，你直接带他走出大楼，去取款机取了500美元塞给他，让他回家陪家人。还有人说，当你不得不裁掉Glitch的员工时，你流下了真诚的眼泪，并花了大量时间帮他们找工作、延长遣散费，对待这件事极其严肃，远超一般CEO。还有人说你支付了员工100%的医保，为了让他们少操心。Slack上市时，你选择了对员工最有利的方式：直接上市，没有锁定期。Slack的收购案也被认为对员工非常友好。对客户也是如此：疫情期间给困难企业免费额度；推出了创新的“公平计费”（Fair Billing），不收那些不活跃账号的钱，即使合同里写了要收。你还经常推迟发布计划，只为了把功能做得更好。最后引用一句话：“Stewart是一位将对员工的责任感内化于心的领导者，他会竭尽所能提供最慷慨的条件。这值得赞颂。”首先，我想向你致敬。遇到这样的领导者很罕见也很励志。我想问：这种慷慨在多大程度上是刻意的战略（比如“这是我们赢的方式，长远来看会有回报”），多大程度上只是你的天性？

[01:20:49] Stewart Butterfield

English:

I think a lot of it is just the way I am as a person and I had wonderful parents who raised me right, but I think there is a little bit of a lesson there and I'm just going to assume people's familiarity with the prisoner's dilemma. The acts of generosity to me are, "Oh, I am demonstrating that I am going to cooperate as we iterate in this game." And if you do that, then people will also cooperate and you both benefit. Whereas if you never really know if the other person is going to defect at the first opportunity, then your best bet is to defect. And so there's a game theoretic aspect, usually in games that are much, much, much more complicated than the prisoner's dilemma. I think one thing I didn't touch on before, but to me was important enough, is that at more than one company all hands, I made everyone in the company repeat this as a chant. It was, "In the long run, the measure of our success will be the amount of value that we create for customers." And I wanted to be super clear and explicit about that because it should be if anything you're doing feels like a little bit shady, a little bit cheating, a little bit maximizing at the wrong moment or taking advantage of a customer or anything like that, definitely shouldn't do it. Because to me, I mean I think it's literally true, but it's also an ethical way to run a business. And it's not just that the ethics are good. It's like there's advantages for you. You're able to attract a better class of employees. If all your employees are ethical, then it's going to be a better place for everyone to work and you're going to be happier and you're going to have fewer internal problems and all that stuff. But I think it really is true that especially in the long run, you can't destroy value for your customers and expect to be successful. You have to actually make their lives better. And you could put effort into pointing it out to them and demonstrating that you have created this value and stuff like that, but there's no substitute for actually having created it. And I think that is incredibly important and that implies a real generosity, whether that's in negotiating terms with an enterprise deal or that's policy decisions. One time that it

blew up in our face was our SLA was like, "For any downtime, you get 100 times your money back." Because from my perspective, it's like if we're down for two minutes, it's like pennies. It doesn't really make any difference. If we're down for 10 hours or something like that, then we have bigger problems than paying back people. Fast-forward, we now have hundreds of millions of dollars in revenue and we've gone public. And shortly after we go public, we have one of the biggest outages we ever had. I don't remember how long it was, but it was many hours. But by the time we got that scale, 100 times the money back for the third of a day that we were down was \$8 million or something like that. It didn't cost us any money because we just gave it to people in the form of credits, but it meant that a bunch of revenue that we had already anticipated for the next quarter wasn't going to show up because people's credits were going to offset what they would've otherwise paid us. And so we definitely changed the terms of service after that because being a public company is a little bit different. But in every other respect, I think they were all really important decisions that were helpful in us becoming successful.

中文翻译:

我想很大程度上是天性，我有很好的父母。但这里面也有个教训，我假设大家都熟悉“囚徒困境”。对我来说，慷慨的行为是在表达：“在这个博弈的迭代中，我选择合作。”如果你这么做，别人也会合作，双方都会受益。反之，如果你总担心对方会背叛，那你最好的策略也是背叛。这有一定的博弈论色彩。我之前没提到的一点是，我在全员大会上让大家重复那句口号：“从长远来看，衡量我们成功的标准将是我们为客户创造的价值。”我想非常明确地表达：如果你做的任何事让你觉得有点阴暗、有点欺骗、或者是在利用客户，那就绝对不要做。这不仅是道德的，也是经营业务的正确方式。它能吸引更优秀的员工，让工作环境更好，减少内部问题。你不能通过损害客户价值来获得成功，你必须真正让他们的生活变得更好。你可以花精力去展示价值，但没有什么能替代“真正创造价值”。这种慷慨体现在合同谈判和政策制定中。有一次这让我们“玩火自焚”：我们的服务等级协议（SLA）规定，任何宕机时间都将获得 100 倍的赔偿。当时我觉得宕机两分钟也就几分钱，没关系。快进到我们上市后，年营收数亿，发生了一次长达数小时的大规模宕机。按 100 倍赔偿，我们要赔出 800 万美元。虽然是以额度（Credit）的形式，但这直接抵消了下个季度的预期收入。在那之后我们确实修改了服务条款，因为上市公司的情况有所不同。但在其他方面，我认为这些决定对我们的成功至关重要。

[01:24:47] Lenny Rachitsky

English:

Was that policy ... It was automatic? You didn't even have to claim it. It was just automatically you get this credit?

中文翻译:

那个政策……是自动的吗？用户甚至不需要申请，就自动获得额度？

[01:24:52] Stewart Butterfield

English:

And the default is you don't have to pay if you let us know. This was, "We will automatically, proactively, preemptively without any input from you ..."

中文翻译:

通常的默认做法是“如果你告诉我们，你可以不付钱”。但我们当时是：“我们将自动、主动、抢先在没有任何用户输入的情况下……”

[01:25:00] Lenny Rachitsky

English:

Too generous.

中文翻译:

太慷慨了。

[01:25:01] Stewart Butterfield

English:

"Apply this credit to your account, and just send you a message that it happened. And by the way, we will do it on the aggregate for downtime, even if the issue didn't affect you as a customer."

中文翻译:

“……将额度应用到你的账户，并给你发个消息。顺便说一下，即使这次宕机没影响到你，我们也会统一发放。”

[01:25:13] Lenny Rachitsky

English:

Wow. Too generous. You found the edge of where you want to be.

中文翻译:

哇，确实太慷慨了。你找到了那个边界。

[01:25:13] Stewart Butterfield

English:

Yeah.

中文翻译:

是的。

[01:25:18] Lenny Rachitsky

English:

What was that mantra again that you had the company chant? I think this is a really nice way to end it.

中文翻译:

你让公司重复的那句口号是什么来着？我觉得用它来结束谈话非常棒。

[01:25:22] Stewart Butterfield

English:

It was, "In the long run, the measure of our success will be the amount of value we create for customers."

中文翻译:

[01:25:28] Lenny Rachitsky

English:

Incredible. I'm just trying to picture the entire team at Slack reciting this mantra.

中文翻译:

太棒了。我正在脑补 Slack 整个团队齐声朗诵这句口号的画面。

[01:25:33] Stewart Butterfield

English:

It was hundreds of people. It felt very like, Kim Jong-Un or Stalin or something like that.

中文翻译:

那是好几百人。感觉有点像……金正恩或者斯大林之类的风格。

[01:25:38] Lenny Rachitsky

English:

Well, on that note, most people don't know this about you, but your actual name when you were born was not Stewart. It was Dharma.

中文翻译:

说到这，大多数人不知道，你出生时的名字其实不叫 Stewart，而是叫 Dharma（达摩/法）。

[01:25:46] Stewart Butterfield

English:

Yeah.

中文翻译:

是的。

[01:25:46] Lenny Rachitsky

English:

And this all makes sense as you learn that.

中文翻译:

了解了这一点后，这一切（你的慷慨和哲学）就都说得通了。

[01:25:49] Stewart Butterfield

English:

Yeah. My name is Dharma Jeremy Butterfield, so my parents named me. And when I was 12, I changed it because I just really wanted to be normal and for some reason I thought Stewart was a normal name. And by the way, you'll notice this now that I said it. Any character except for Stuart Little the mouse, anytime you see a character in a movie, a novel, TV show or whatever, there's only the loser Stewart and the asshole Stewart. It's obviously, in the collective consciousness, a terrible name and I shouldn't have chosen it and I regret it. But by the time I realized that, Dharma and Greg had already come out and it would've seemed like I was bandwagon jumping. And people thought it was a girl's name, even though in India it's obviously only a boy's name. I'm going to add just one last little tidbit because I forgot about this earlier on and I think it helps tie things together, and it's called the owner's delusion. And this is based on something I posted on Twitter. The person who came up with the name later deleted their account and so I have no idea who it was and who to credit for this. But what I had posted was, and this was a long time ago when restaurant websites have gotten better and it doesn't really matter because Google Local was taking over everything, but this is, let's say, 10 years ago. There's five things you could possibly want when you go to a restaurant's website and it's their street address, their phone number, the menu, the hours of operation ... Oh my God, I forgetting the fifth thing. Oh, and making a reservation, how to make a reservation. And again, this problem has to some extent taken care of it's itself or at least improved, but what you would get was this super slow loading photo, the Ken Burns effect as it [inaudible 01:27:30] ...

中文翻译:

是的，我出生时的名字是 Dharma Jeremy Butterfield。12 岁时我改了名，因为我当时只想做一个“正常人”，不知为何我觉得 Stewart 是个正常的名字。顺便说一下，你会发现除了《精灵鼠小弟》(Stuart Little)，电影或小说里的 Stewart 要么是失败者，要么是混蛋。这显然是个糟糕的名字，我很后悔。但当我意识到这一点时，美剧《Dharma & Greg》已经火了，改回去就像在蹭热度。而且人们总觉得 Dharma 是女孩名，虽然在印度这显然是男孩名。最后我再分享一个细节，叫“所有者错觉”(Owner's Delusion)。这是基于我很久以前发的一条推特。当时餐厅网站做得很烂：你去餐厅网站只想知道五件事：地址、电话、菜单、营业时间、如何预订。但你看到的往往是加载极慢的大图、华丽的转场特效……

[01:27:29] Lenny Rachitsky

English:

The flashed.

中文翻译:

还有那些 Flash 动画。

[01:27:30] Stewart Butterfield

English:

And then fading in and then some music starts playing. And then if they show you the phone number, it's not clickable.

中文翻译:

然后是淡入淡出，接着开始放音乐。好不容易看到电话号码，结果还不能点击拨号。

[01:27:38] Lenny Rachitsky

English:

Image.

中文翻译:

因为那是张图片。

[01:27:39] Stewart Butterfield

English:

It's not even text that you can copy because yes, it's an image. And they don't have the hours. They don't put the address or whatever and it's just like, "What?" For sure, whoever made this website for the restaurant owner and the restaurant owner themselves have definitely been in the position where they went to somebody else's restaurant website because they wanted to get the address or the opening hours or the phone number or whatever. So why does it end up like this and what should we call this? And whoever replied to the tweet, she said, "We should call it the owner's delusion," and I was like, "Oh my God. That's perfect." And I think that is incredibly powerful and what ends up with the result, like Apple naming whatever that feature is called Sleep, which it's too hard to understand what that can possibly mean. And that's why people anticipate, despite the fact that when they get to your website for the first time, their intent is absolutely the minimum number of micro points above the threshold required from them to actually take that action. You're like, "All right. Welcome to my website," and there's a bunch of BS and there's a bunch of stuff that doesn't make any sense and the buttons are inscrutable. And it's unclear what to do next because I think that my thing is so important and I don't recognize that you are at work and you were late this morning and you have to go to the bathroom and you're just a regular human being who has stuff going on, that you're concerned that your kid is a fuck-up and they're getting in trouble at school and stuff like that. They're not subjects who paid money to go to your play and are sitting in the audience and waiting for that curtain to go out. They're people who are going to bounce in a fraction of a second. And so everyone should always be conscious of the owner's solution.

中文翻译:

甚至连复制都不行，因为那是张图片。他们不放营业时间，不放地址。这太离谱了。餐厅老板和网页设计师肯定也遇到过想找别家餐厅地址却找不到的情况。为什么会变成这样？推特上有人回复说：“这叫‘所有者错觉’。”我觉得太精辟了。这就是为什么Apple会把功能命名为“睡眠”这种让人摸不着头脑的名字。老板总觉得自己的产品极其重要，却忘了用户只是个普通人：他们可能上班迟到了，可能急着上厕所，可能在担心孩子在学校闯祸。他们不是花钱买票坐着等你看戏的观众，他们是随时会关掉页面的普通人。所以每个人都应该时刻警惕这种“所有者错觉”。

[01:29:27] Lenny Rachitsky

English:

I love that. What's the solution? Is it have other people look at it and give you feedback?

中文翻译:

太棒了。那解决方案是什么？是让别人来看并给你反馈吗？

[01:29:31] Stewart Butterfield

English:

Yeah, and recognize it. And unfortunately, it's one of those things like Murphy's Law.

中文翻译:

是的，并且要意识到它的存在。不幸的是，这就像墨菲定律一样。

[01:29:35] Lenny Rachitsky

English:

Yeah.

中文翻译:

是的。

[01:29:37] Stewart Butterfield

English:

Even you can go wrong even when you take into account Murphy's Law.

中文翻译:

即使你考虑到了墨菲定律，事情依然可能出错。

[01:29:39] Lenny Rachitsky

English:

That's right.

中文翻译:

没错。

[01:29:41] Stewart Butterfield

English:

But if you don't name it and recognize it and discuss it and train yourself to think that way, take a breath, pretend you're a regular person, and then look at this again and see if it makes sense, then you're screwed.

中文翻译:

但如果你不给它命名、不承认它、不讨论它，不训练自己换位思考（深呼吸，假装自己是个普通人，再看一遍产品），那你就完蛋了。

[01:29:55] Lenny Rachitsky

English:

I love that. I love that you threw this in here. I have a billion other questions I'm going to ask you in part two when we do this someday. Stewart, thank you so much for doing this. Thank you so much for being here.

中文翻译:

我太喜欢这个观点了。很高兴你分享了这些。我还有无数个问题想问你，留到以后的第二部分吧。Stewart，非常感谢你参加节目，非常感谢你的到来。

[01:30:04] Stewart Butterfield

English:

Yeah. Thank you for having me, Lenny. I really enjoyed it.

中文翻译:

谢谢你的邀请，Lenny。我非常享受这次谈话。

[01:30:07] Lenny Rachitsky

English:

Same here. Bye, everyone. 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. See you in the next episode.

中文翻译:

我也是。大家再见。非常感谢大家的收听。如果你觉得这期节目有价值，可以在 Apple Podcasts、Spotify 或你喜欢的播客应用中订阅。此外，请考虑给我们评分或留下评论，这能极大地帮助其他听众发现这个播客。你可以在 lennyspodcast.com 找到所有往期节目或了解更多信息。下期节目再见。