AMJAD MASAD

LENNY'S PODCAST

BILINGUAL TRANSCRIPT

ORIGINAL BY

Lenny Rachitsky

@lennysan • x.com/lennysan

ANALYSIS BY

@Penny777 • x.com/penny777

Generated: 2026-01-16

Amjad Masad - 双语对照

This is the complete bilingual transcript for Lenny's Podcast featuring Amjad Masad, the co-founder and CEO of Replit.

[00:00:00] Amjad Masad

English:

The idea behind Replit is that making software today is very difficult. We want to make it easier. People view this as a developer in their pocket essentially. We have 34 million users globally. There's people everywhere learning to code on Replit, building startups, building personal software, personal tools.

中文翻译:

Replit 背后的理念是：在当今时代，开发软件非常困难。我们希望让它变得更简单。人们基本上把它看作是“口袋里的开发者”。我们在全球拥有 3400 万用户。世界各地都有人在 Replit 上学习编程、创建初创公司、开发个人软件和个人工具。

[00:00:20] Lenny Rachitsky

English:

For people building products, say, product managers, founders, what skills do you see will matter more, matter less?

中文翻译:

对于那些构建产品的人，比如产品经理、创始人，你认为哪些技能会变得更重要，哪些会变得不那么重要?

[00:00:27] Amjad Masad

English:

Typically, you're bottlenecked where your ideas are not fitting in because they need to be made and they need to be made quickly. Now, you open up that bottleneck. So now actually making things is a lot easier. Actually, you become limited by how fast you can generate ideas.

中文翻译:

通常情况下，你的瓶颈在于想法无法落地，因为它们需要被制造出来，而且需要快速制造。现在，你打破了这个瓶颈。所以现在实际动手做东西变得容易得多。事实上，你现在的限制变成了你产生想法的速度。

[00:00:44] Lenny Rachitsky

English:

I think people are unaware of just how far things have gone.

中文翻译:

我认为人们还没有意识到事情已经发展到了什么程度。

[00:00:47] Amjad Masad

English:

I could imagine whatever five years from now, someone running a billion-dollar company with zero employees where it's like the support is handled by AI, the development is handled by AI, and you're just building and creating this thing.

中文翻译:

我可以想象，五年后的某一天，有人经营着一家估值十亿美元的公司，却没有任何员工。客户支持由 AI 处理，开发由 AI 处理，而你只是在构建和创造这个东西。

[00:01:01] Lenny Rachitsky

English:

Man, the future is wild. Today, my guest is Amjad Masad. Amjad is the co-founder of Replit, an AI-powered software development and deployment platform for building and shipping software. It's one of the fastest-growing developer communities and AI products in the world. There's a lot of talk these days about how AI is changing, how products will be built, how product teams are going to operate, which functions will be more and less valuable over time. But I feel like very few people have actually seen what modern AI tools can do and have fully grasped how much you can get done with very little technical skill now and in the future. And so I'm going to do an experiment with this podcast where I'm going to do a series of behind the product episodes where we go deep on important products that product builders should be aware of and should probably start playing with.

(00:01:54):

In our conversation, Amjad does a demo of what you can do with Replit today, which is going to blow your mind. And then we spend most of the conversation talking about the implications of this on the future of product development, on the future of product management, and on the future of startups and founders. It's a very exciting time. It's also a very scary and destabilizing time for a lot of people, and my thinking is the more you are aware of what's possible today and where things are going, the better position you'll be in to thrive in this very wild and crazy future that is coming very fast. If you enjoy this podcast, don't forget to subscribe and follow it in your favorite podcasting app or YouTube. It's the best way to avoid missing feature episodes, and it helps the podcast tremendously. With that, I bring you Amjad Masad. Amjad, thank you so much for being here. Welcome to the podcast.

中文翻译:

伙计，未来真是太疯狂了。今天，我的嘉宾是 Amjad Masad。Amjad 是 Replit 的联合创始人。Replit 是一个由 AI 驱动的软件开发和部署平台，用于构建和发布软件。它是全球增长最快的开发者社区和 AI 产品之一。最近有很多关于 AI 如何改变产品构建方式、产品团队如何运作、以及随着时间的推移哪些职能会变得更有价值或更没价值的讨论。但我觉得很少有人真正见过现代 AI 工具能做什么，也很少有人完全理解现在和未来，即使技术技能非常有限，也能完成多少工作。因此，我打算在这个播客中做一个实验，我会做一系列“产品背后”的剧集，深入探讨产品构建者应该了解并可能应该开始尝试的重要产品。

(00:01:54):

在我们的对话中，Amjad 演示了你今天可以用 Replit 做什么，这会让你大吃一惊。然后，我们大部分时间都在讨论这对产品开发的未来、产品管理的未来以及初创公司和创始人的未来的影响。这是一个非常令人兴奋的时代。对很多人来说，这也是一个非常可怕和动荡的时代。我的想法是，你越了解今天什么是可能的以及事情的发展方向，你就越能在即将到来的疯狂未来中占据有利地位。如果你喜欢这个播客，别忘了在你不常用的播客应用或 YouTube 上订阅和关注。这是避免错过未来节目的最好方式，也对播客有很大帮助。那么，现在有请 Amjad Masad。Amjad，非常感谢你能来。欢迎来到播客。

[00:02:46] Amjad Masad

English:

It's my pleasure.

中文翻译:

这是我的荣幸。

[00:02:47] Lenny Rachitsky

English:

I thought it'd be great to start with just having you explain what is Replit? What's the vision? Where is this going? What job does it do for people?

中文翻译:

我想先请你解释一下什么是 Replit？它的愿景是什么？它将走向何方？它为人们解决了什么问题？

[00:02:56] Amjad Masad

English:

The idea behind Replit is that making software today is very difficult and we want to make it easier. One of the reasons for the difficulty is that it is very fragmented, so you would need to download what's called an IDE. It's basically a code editor. You need to download the runtime, basically Python or JavaScript, need to figure out a package manager to configure your kind of open source packages, and once you've done all of that, you need to figure out how to deploy it, how to share it. So it's a very hard process, and that's one of the ways where people get stuck and never learn how to code because it just feels like this cumbersome IT process.

(00:03:42):

And so the vision for Replit has always been is like, okay, making software is fun, is great, more people should do it, but so for more people to do it, it needs to be easier to do, it needs to be in one place, and it needs to be learnable. It's easier to learn. So that's the product today. It is I think one of the more easier IDEs/ environment/deployment environment on the internet, and I think we make it really easy for people to just jump in even without prior experience of coding, especially now with the new AI products that we built.

中文翻译:

Replit 背后的想法是，现在的软件开发非常困难，我们想让它变得更简单。困难的原因之一是它非常碎片化，你需要下载所谓的 IDE（集成开发环境），基本上就是一个代码编辑器。你需要下载运行时（Runtime），比如

Python 或 JavaScript，需要搞清楚包管理器（Package Manager）来配置你的开源包，一旦你完成了所有这些，你还得搞清楚如何部署它，如何分享它。所以这是一个非常艰难的过程，这也是人们卡住并永远学不会编程的原因之一，因为它感觉就像是一个繁琐的 IT 流程。

(00:03:42):

所以 Replit 的愿景一直是：开发软件很有趣，很棒，应该有更多人参与进来。但为了让更多人参与，它必须变得更容易，必须集成在一个地方，而且必须是可学习的。它更容易上手。这就是今天的产品。我认为它是互联网上最简单的 IDE/环境/部署环境之一，而且我认为我们让人们即使没有编程经验也能轻松加入，尤其是现在有了我们构建的新 AI 产品。

[00:04:22] Lenny Rachitsky (Ad Break)

English:

This episode is brought to you by WorkOS. If you're building a SaaS app, at some point, your customers will start asking for enterprise features like SAML authentication and SCIM provisioning. That's where WorkOS comes in, making it fast and painless to add enterprise features to your app. Their APIs are easy to understand so that you can ship quickly and get back to building other features. Today, hundreds of companies are already powered by WorkOS, including ones you probably know like Vercel, Webflow, and Loom. WorkOS also recently acquired Warrant, the Fine Grained Authorization service. Warrant's product is based on a groundbreaking authorization system called Zanzibar, which was originally designed for Google to power Google Docs and YouTube. This enables fast authorization checks at enormous scale while maintaining a flexible model that can be adapted to even the most complex use cases. If you're currently looking to build role-based access control or other enterprise features like single sign-on, SCIM, or user management, you should consider WorkOS. It's a drop-in replacement for Auth0 and supports up to 1 million monthly active users for free. Check it out at workos.com to learn more. That's workos.com.

(00:05:40):

This episode is brought to you by Persona, the adaptable identity platform that helps businesses fight fraud, meet compliance requirements, and build trust. While you're listening to this right now, how do you know that you're really listening to me, Lenny? These days, it's easier than ever for fraudsters to steal PII, faces, and identities. That's where Persona comes in. Persona helps leading companies like LinkedIn, Etsy, and Twilio securely verify individuals and businesses across the world. What sets Persona apart is its configurability. Every company has different needs depending on its industry, use cases, risk tolerance, and user demographics. That's why Persona offers flexible building blocks that allow you to build tailored collection and verification flows that maximize conversion while minimizing risk. Plus Persona's orchestration tools automate your identity process so that you can fight rapidly shifting fraud and meet new waves of regulation. Whether you're a startup or an enterprise business, Persona has a plan for you. Learn more at withpersona.com/lenny. Again, that's withpersona.com/lenny.

中文翻译:

本期节目由 WorkOS 赞助。如果你正在构建 SaaS 应用，在某些时候，你的客户会开始要求企业级功能，如 SAML 身份验证和 SCIM 配置。这就是 WorkOS 的用武之地，它能让你快速、无痛地为应用添加企业级功能。他们的 API 易于理解，因此你可以快速发布并回到其他功能的开发中。今天，数百家公司已经由 WorkOS 提供支持，包括你可能熟悉的 Vercel、Webflow 和 Loom。WorkOS 最近还收购了细粒度授权服务 Warrant。Warrant 的产品基于一个名为 Zanzibar 的开创性授权系统，该系统最初是为 Google 设计的，用于支持 Google Docs 和 YouTube。这使得在大规模情况下进行快速授权检查成为可能，同时保持了可以适应最复杂用例的灵活模型。如果你目前正在寻求构建基于角色的访问控制或其他企业功能（如单点登录、SCIM 或用户管理），你应该考虑 WorkOS。它是 Auth0 的直接替代品，并免费支持多达 100 万月活跃用户。访问 workos.com 了解更多信息。

(00:05:40):

本期节目由 Persona 赞助，这是一个适应性强的身份平台，可帮助企业打击欺诈、满足合规要求并建立信任。当你现在听这段音频时，你怎么知道你真的在听我——Lenny 的声音？如今，欺诈者比以往任何时候都更容易窃取个人身份信息（PII）、面部信息和身份。这就是 Persona 的用武之地。Persona 帮助 LinkedIn、Etsy 和 Twilio 等领先公司安全地验证全球范围内的个人和企业。Persona 的独特之处在于其可配置性。每家公司根据其行业、用例、风险承受能力和用户人口统计数据都有不同的需求。这就是为什么 Persona 提供灵活的构建块，允许你构建量身定制的收集和验证流程，在最大限度提高转化率的同时最大限度降低风险。此外，Persona 的编排工具可自动执行你的身份流程，以便你可以应对快速变化的欺诈并满足新的监管浪潮。无论你是初创公司还是企业，Persona 都有适合你的方案。在 withpersona.com/lenny 了解更多信息。

[00:06:55] Lenny Rachitsky

English:

What's the scale of Replit at this point? How large has this gotten? How many people are using it?

中文翻译:

Replit 目前的规模有多大？它发展到什么程度了？有多少人在使用它？

[00:06:56] Amjad Masad

English:

We have 34 million users globally. We have a large global presence. There's people everywhere learning to code on Replit, building startups, building personal software, personal tools or internal tools of the companies. More recently, we've been expanding to companies. We released our kind of B2B package in July, and that's been growing really fast. It's been really fun to see people bring Replit to work as well.

中文翻译:

我们在全球拥有 3400 万用户。我们的全球影响力很大。世界各地都有人在 Replit 上学习编程、创建初创公司、开发个人软件、个人工具或公司的内部工具。最近，我们一直在向企业端扩展。我们在 7 月发布了 B2B 软件包，增长非常快。看到人们把 Replit 带到工作中也很有趣。

[00:07:25] Lenny Rachitsky

English:

Damn, I knew it was popular. I didn't realize it was that large actually. As I was preparing for this podcast episode, there's this tweet that went viral where this guy, Jevin, who I actually know. I know this guy from Canada, he's awesome, tweeted about how his 11-year-old girl built an app in Replit. She just had an idea and she built it. And the best part of it is someone in the... replied to him and they're like, "You have to launch an app. You have to host it somewhere. You have to build a database. You have to deploy it. There's no way to do that." And he's like, "No, that's exactly what Replit did."

中文翻译:

天哪，我知道它很受欢迎，但没意识到规模已经这么大了。在我准备这期播客时，有一条推文疯传，是一个叫 Jevin 的人发的，我其实认识他。他来自加拿大，人很棒。他发推说他 11 岁的女儿在 Replit 上做了一个应用。她只是有个想法，然后就把它做出来了。最精彩的部分是有人回复他说：“你得发布应用，得找地方托管，得建数据库，得部署。不可能就这么做出来。”结果 Jevin 回复说：“不，这正是 Replit 所做的事情。”

[00:07:59] Amjad Masad

English:

Yeah, that's what we do. Everything that commenter was talking about, and he's right. The surprising thing about an 11-year-old building an app is not so much even the coding, it is like all the nonsense around it, and so we just abstract all that away.

中文翻译:

是的，这就是我们做的。那个评论者说的所有事情（都是传统开发需要的），他是对的。一个 11 岁孩子能做出应用，令人惊讶的其实不在于编程本身，而在于处理周围那些乱七八糟的琐事，而我们将所有这些都抽象化（简化）了。

[00:08:19] Lenny Rachitsky

English:

I love that and I struggled with that myself when I was an engineer way back in the day. Oh, you were an engineer. I didn't know that. I was. I was an engineer for 10 years. I was an engineering manager, and then I jumped ship into product.

中文翻译:

我太喜欢这一点了，我以前当工程师的时候也深受其苦。噢，你以前是工程师？我以前不知道。是的，我当了 10 年工程师，还做过工程经理，然后才转行做产品。

[00:08:32] Amjad Masad

English:

Wow.

中文翻译:

哇。

[00:08:32] Lenny Rachitsky

English:

I'm happy I did but I do miss that. I was not an amazing engineer. I was a good enough startup engineer, so this is the kind of stuff that I would've left to use. So we're going to jump into a demo of what this actually looks like. I thought maybe actually before we get into it, there's other tools that people are aware of that help you build stuff. And so to put a finer point on what this does and how it's different from other things, you may have heard of, say, Cursor, it comes up a lot these days, just talk about a little bit about the competitive landscape of who else is out there that helps you build product.

中文翻译:

我很庆幸自己转行了，但我确实很怀念那段时光。我不是那种顶尖的工程师，但我是一个足够好的初创公司工程师，所以这类工具正是我梦寐以求的。接下来我们要进入演示环节，看看它到底长什么样。我想在开始之前，大家可能还知道其他一些帮助构建产品的工具。为了更准确地说明 Replit 的功能以及它与其他工具的区别——比如你可能听说过 Cursor，最近它经常被提起——请谈谈目前的竞争格局，还有谁在帮助人们构建产品。

[00:09:04] Amjad Masad

English:

Again, we go back to this idea of end-to-end platform for making software. So that's from writing code all the way to deploying it, and monetizing it and all of that. Now, every step in the process of the software development lifecycle, there are a lot of different tools. So Cursor is a fork of VS Code that's made that has really awesome AI tools, but that's an editor. You still need runtime, you still need a deployment environment. Actually, quite a few users use Cursor in tandem with Replit, because Replit just simplifies the runtime and deployment environment.

(00:09:42):

And so you have products, AI products, different places in the software development lifecycle, but really what differentiates Replit is that we do everything, but also that makes it harder to adopt for certain people. If you're at a big company, it's very easy to bring a new editor and start coding with it. It's quite hard to bring something that's quite opinionated about everything from how the code runs to how the code deploys, but that's the trade-off we're willing to make is like yeah, we're not going to get into the enterprise main software development pipeline, but we want to empower everyone to be able to build software, and that means product managers, designers. We have operations people, sales ops, HR ops. We have lawyers using Replit, and so it is democratizing the act of software engineering.

中文翻译:

我们再次回到“端到端软件制作平台”这个理念。这意味着从编写代码一直到部署、货币化等等。现在，在软件开发生命周期的每一步，都有很多不同的工具。比如 Cursor 是 VS Code 的一个分支，它拥有非常棒的 AI 工具，但它只是一个编辑器。你仍然需要运行时，仍然需要部署环境。实际上，相当多的用户将 Cursor 与 Replit 配合使用，因为 Replit 简化了运行时和部署环境。

(00:09:42):

所以你在软件开发生命周期的不同阶段都有 AI 产品，但 Replit 的真正区别在于我们涵盖了所有环节。当然，这也使得某些人更难采用它。如果你在一家大公司，引入一个新的编辑器并开始编码很容易。但要引入一个对从代码运行到部署的所有环节都有“主见”（Opinionated）的工具就很难了。但这是我们愿意做的权衡：是的，我们可能不会进入企业的主流软件开发流水线，但我们想让每个人都能构建软件，这包括产品经理、设计师。我们有运营人员、销售运营、人力资源运营，甚至有律师在使用 Replit。所以，它正在使软件工程行为民主化。

[00:10:49] Lenny Rachitsky

English:

Amazing, and that's why you're here. Let's do a demo. While you're pulling it up, you're going to share your screen and show us what this product can do. And the reason I am excited about doing a demo, and this is an experiment, kind of a new type of podcast episode I'm doing where we're diving into specific products and what they can do, I feel like there's so much talk about AI and what it's doing and people keep reading about, oh, AI can do this and AI can do that, and I feel like not many people actually see this stuff in action, especially the most cutting edge stuff. I think people are unaware of just how far things have gone and how much is actually possible, especially when someone that knows what they're doing is using the product. So I'm excited to show people what is actually possible and especially because this is going to impact the future of product management and product teams. So I'll turn that over to you. Give us a demo.

中文翻译:

太棒了，这就是你来这里的原因。让我们开始演示吧。在你准备的时候，你可以分享屏幕，向我们展示这个产品能做什么。我之所以对演示感到兴奋，是因为这是一个实验，是我正在尝试的一种新型播客形式，我们会深入探讨具体的产品及其功能。我觉得现在关于 AI 的讨论太多了，人们一直在读到“AI 能做这个”、“AI 能做那个”，但我觉得没多少人真正见过这些东西的实际运作，尤其是最前沿的技术。我认为人们没有意识到事情已经发展到了什么程度，以及现在到底能实现多少东西，特别是当一个懂行的人在使用这个产品时。所以我很高兴能向大家展示什么是真正可能的，尤其是因为这将影响产品管理和产品团队的未来。那么，交给你了，请开始演示。

[00:11:37] Amjad Masad

English:

Awesome. So this is Replit's homepage. You can create what's called a Repl, which is a project. We have all sorts of languages. You can pick from really in the hundreds, but most recently, and this is how Replit became a thousand times easier, is you can just describe what you want to make. So you go on this home page, we have this text box, and you can write something like make me a cool app or what have you, but a more descriptive prompt is better.

(00:12:11):

And so I asked RPM at Replit, Aman Mathur who's a fan of the show to tell me what PMs like to build. And so he came up with a prompt. He kind of really crafted a great prompt. So I'm going to put it here. And basically, what we're asking for is we want to build a web application. You can say what stack you want to use or you can leave it up to the AI to decide. Here we're saying build it in Node.js for product managers to track feature requests on a public dashboard. So say I have a product, I'm growing, I have a community, I want that community to engage with building the product. I want them to submit feature requests, vote on them. I want to be able to manage that. So we're talking here about the features of voting system, feature requests.

中文翻译:

太好了。这是 Replit 的主页。你可以创建一个所谓的 Repl，也就是一个项目。我们支持各种语言，你可以从数百种语言中选择。但最近，Replit 变得简单了一千倍，因为你只需要描述你想做什么。在主页上，我们有一个文本框，你可以写类似“给我做一个酷炫的应用”之类的话，但描述性更强的提示词（Prompt）效果更好。

(00:12:11):

所以我问了 Replit 的产品经理 Aman Mathur（他也是这个节目的粉丝），问他 PM 们喜欢构建什么。他想出了一个提示词，写得非常好。我现在把它放进去。基本上，我们的要求是构建一个 Web 应用程序。你可以指定使用什么技术栈，也可以让 AI 决定。这里我们说的是用 Node.js 构建一个供产品经理在公共仪表板上跟踪功能请求的应用。假设我有一个产品，正在成长，我有一个社区，我希望社区参与到产品的构建中。我希望他们提交功能请求并进行投票，而我能够管理这些。所以我们这里讨论的功能包括投票系统和功能请求。

[00:13:02] Lenny Rachitsky

English:

Read a few of them just for folks that aren't watching on YouTube to give them, send some of the stuff in this prompt.

中文翻译:

为那些没在 YouTube 上看视频的听众读几个功能点，让他们感受一下提示词里的内容。

[00:13:07] Amjad Masad

English:

So a feature requests submission, so allowing the users to add features. A voting system, so allowing users for these features, feature requests and status tracking, being able to, it's like a kind of advance style board with columns like planned and progress, so that way the admin can kind of share with the community what they're building. And we want it to be user-friendly design, so make it modern and all of that nice kind of prompty things. And then admin controls for product manager. So as a product manager, I want to be able to really manage this community.

中文翻译:

好的，包括：功能请求提交，允许用户添加功能；投票系统，允许用户对这些功能请求进行投票；状态跟踪，能够像看板一样，有“计划中”、“进行中”等列，这样管理员就可以向社区分享他们正在构建的内容。我们还要求用户友好的设计，使其看起来现代感十足，以及所有那些提示词里常见的修饰语。最后是产品经理的管理员控制台。作为 PM，我希望能真正管理这个社区。

[00:13:44] Lenny Rachitsky

English:

I love that it builds internal tools too, not just the front end.

中文翻译:

我喜欢它不仅能构建前端，还能构建内部工具。

[00:13:47] Amjad Masad

English:

Yes. Exactly. Exactly. All right, so we're going to start building. Since this is a pretty big, big prompt, the initial coding might take a while. There's different styles of using Replit agents. I often go with minimalist prompts. That's also how I code as well. I have a vague idea for what I want to build and iterate from there. Other people, product managers like to write PRDs and more descriptive things, and you can do either of those things. The AI now responded and then said, I'll build all of that for you. I'm going to build up the initial prototype and you can tell me how it feels, and then we can make it better from there. The AI is also suggesting, adding comment threads, implementing email notifications, and so I can select those and it's being creative, it's telling me what else I could build, but for now, I'm just going to go with a prototype and then we can assess from there.

(00:14:50):

So as you see, as the prototype is starting, you can see this progress pane where we can watch the AI doing its thing here. It's created a Postgres database. Obviously, when we're building a full-stack application, you need to be able to save things. This is one of the cool things about Replit. We have all these services, storage, database. So now it's coding, it's building the database schema. Now, it's building the home page, and it's actually quite fun and edifying to watch it build this, because you can really start to learn how to structure web apps. And if it runs into a problem and as things get complicated, it might run into a problem and you want to be able to help debug and things like that, it's good to be able to have an idea of what's going on, but it's not necessary. I think a lot of people just don't care about the code and are still able to build things, but we want to make the process transparent. We want to show people exactly what the agent is doing.

中文翻译:

是的，没错。好了，我们要开始构建了。由于这是一个相当大的提示词，初始编码可能需要一点时间。使用 Replit Agent 有不同的风格。我通常喜欢用极简的提示词，这也是我编程的方式：我有一个模糊的想法，然后从那里开始迭代。其他人，比如产品经理，喜欢写 PRD（产品需求文档）和更详细的描述，这两种方式都可以。AI 现在回应了，说：“我会为你构建所有这些。我会先建立初始原型，你可以告诉我感觉如何，然后我们再改进。” AI 还建议添加评论线程、实现邮件通知等，我可以选择这些。它很有创意，会告诉我还能构建什么，但目前我先只做原型，然后再评估。

(00:14:50):

如你所见，原型开始构建了，你可以看到这个进度面板，我们可以观察 AI 在这里操作。它创建了一个 Postgres 数据库。显然，构建全栈应用时，你需要能够保存数据。这是 Replit 的酷炫之处之一：我们拥有所有这些服务，包括存储和数据库。现在它正在编码，构建数据库模式（Schema）。现在它正在构建主页。看着它构建这些其实很有趣，也很有启发性，因为你可以真正开始学习如何构建 Web 应用。如果它遇到问题（随着事情变得复杂，它可能会遇到问题），而你想帮忙调试之类的，了解发生了什么会很有帮助，但这不是必须的。我认为很多人根本不在乎代码，但仍然能做出东西，但我们希望过程透明，向人们展示 Agent 到底在做什么。

[00:15:57] Lenny Rachitsky

English:

You're basically sitting there behind an engineer on a computer and just watching them code is what the experience feels like.

中文翻译:

这种体验感觉就像你坐在一个工程师后面，看着他在电脑上写代码。

[00:16:04] Amjad Masad

English:

Yeah, and actually, the way we built it is it's a multiplayer system. So Replit has real time, what we call multiplier coding, and we reused the multiplayer system to build the agent. So the agent in the code is structured as another user of the platform. So basically, we're both coding together. So I can go into the files here and that's the thing that makes Replit really cool. I think people are familiar with some of the more chat interfaces like v0 and others where it's purely chat, but this is a full IDE where you can go and look at the files and edit them yourself or ask the AI for an explanation.

中文翻译:

是的，实际上我们的构建方式是一个“多玩家”系统。Replit 拥有实时的、我们称之为“多人协作编码”的功能，我们复用了这个系统来构建 Agent。所以代码中的 Agent 被结构化为平台的另一个用户。基本上，我们是在一起编码。我可以进入这里的文件，这就是 Replit 真正酷的地方。我想人们可能熟悉一些纯聊天的界面，比如 v0 之类的，但这是一个完整的 IDE，你可以查看文件，自己编辑，或者要求 AI 解释。

[00:16:51] Lenny Rachitsky

English:

What's the limitation of what this can do today? What can't you do? Say you're like, you have zero coding experience, what sorts of products can you not yet build with something like this that might be possible in the future? How far does this take you now?

中文翻译:

目前它的局限性是什么？有什么是你做不到的？假设你完全没有编程经验，有哪些产品是目前用这种工具还做不出来，但未来可能实现的？它现在能帮你走到哪一步？

[00:17:08] Amjad Masad

English:

You can build MVPs. I think you can also start to get some initial users. I think when you start iterating on the product like large iterations, you might run into problems. For example, it's not very good at database migrations, and so we're trying to fix that. So when you're iterating on the product, a lot of the times, you're actually changing the structure of the app and that requires database migrations. And so now it might change the database in a way that creates an error that's unrecoverable. And at that point, you might get stuck, especially if you don't know how to code.

(00:17:56):

Some people will figure it out by going to ChatGPT and Claude and asking questions and I actually am really inspired about how persistent some of our users are, which is really amazing. But I think you'll get an MVP pass, the MVP where it's a product that's working and you need to change and iterate on it. It's still a struggle now, but I expect over the next few months, we'll continue. It's if you think about it's like we're building as you were building, so we're building out the agent so that it can continue getting better as our users are also building their applications.

中文翻译:

你可以构建 MVP（最小可行性产品）。我认为你也可以开始获取一些初始用户。但当你开始对产品进行大规模迭代时，可能会遇到问题。例如，它目前不太擅长数据库迁移（Database Migrations），我们正在努力修复这个问题。当你迭代产品时，很多时候你实际上是在改变应用的结构，这需要数据库迁移。现在它可能会以某种方式更改数据库，从而导致无法恢复的错误。在那一点上，你可能会卡住，尤其是如果你不懂编程的话。

(00:17:56):

有些人会通过去问 ChatGPT 或 Claude 来解决问题，我其实被一些用户的执着深深打动了，这真的很了不起。但我认为你能做出一个 MVP，但过了 MVP 阶段，当它成为一个正在运行的产品且你需要不断更改和迭代时，现在仍然有些吃力。但我预计在接下来的几个月里，我们会继续改进。如果你想一下，这就像我们在你构建的同时也在构建——我们正在完善 Agent，以便随着用户构建应用，它也能变得越来越好。

[00:18:40] Lenny Rachitsky

English:

Got it. So what I'm hearing is it's really good at building the first version and helping you get to something that you can even have people use. It's not amazing yet evolving from there, using AI to help you make the product better and better and better and iterate.

中文翻译:

明白了。所以我听到的是，它非常擅长构建第一个版本，并帮你做出一个甚至可以让别人使用的东西。但在那之后的演进，即利用 AI 让产品变得越来越好并不断迭代方面，目前还不够完美。

[00:18:55] Amjad Masad

English:

Yes.

中文翻译:

是的。

[00:18:55] Lenny Rachitsky

English:

But you can get in there if you know how to code and take it from there, right?

中文翻译:

但如果你懂编程，你可以自己进去接手，对吧?

[00:19:00] Amjad Masad

English:

Yes or you can hire someone. We have a feature on the site called Bounties where you can hire human coders to kind of help you finish.

中文翻译:

是的，或者你可以雇人。我们网站上有一个叫 Bounties（悬赏）的功能，你可以雇佣人类程序员来帮你完成。

[00:19:14] Lenny Rachitsky

English:

That's going to be our job for humans. That'll remain for a while.

中文翻译:

那是我们人类的工作。这还会持续一段时间。

[00:19:18] Amjad Masad

English:

You know what we want to do? We want to get to a point where the agent can go grab a human when it runs into a problem. I think that would be sick.

中文翻译:

你知道我们想做什么吗？我们想达到这样一个阶段：当 Agent 遇到问题时，它可以主动去“抓”一个人类来帮忙。我觉得那会非常酷。

[00:19:32] Lenny Rachitsky

English:

Oh, my God. It's like everything's reversed. I love it. Oh, I think it might be done. Check that out.

中文翻译:

噢，天哪。就像一切都反过来了。我喜欢这个主意。噢，我想它可能做好了。快看。

[00:19:38] Amjad Masad

English:

Yeah. So now the agent is asking us, is the application running and showing the homepage?

中文翻译:

是的。现在 Agent 在问我们：应用程序是否正在运行并显示主页?

[00:19:46] Lenny Rachitsky

English:

Like it's confirming.

中文翻译:

就像是在确认。

[00:19:48] Amjad Masad

English:

Yeah, almost asking us to do a QA. I'll just say yes. So it found an error. So there's an error here and it's like there's a dumb warning, "I'm going to fix it." So in the meantime, as it's fixing it, so it can be proactive, right? Because it looks at all the errors and things like that, but in the meantime, we can use it. I just created an account. It's coding.

中文翻译:

是的，几乎是在要求我们做 QA（质量保证）。我就说“是”。哦，它发现了一个错误。这里有一个错误，它说有一个愚蠢的警告，“我要修复它”。与此同时，在它修复的时候——它是可以主动出击的，对吧？因为它会查看所有的错误之类的——与此同时，我们可以使用它。我刚刚创建了一个账户。它还在编码。

[00:20:12] Lenny Rachitsky

English:

[inaudible] It's cool.

中文翻译:

（听不清）这很酷。

[00:20:11] Amjad Masad

English:

Let's see how it restart. Okay, we'll wait for it.

中文翻译:

看看它怎么重启。好，我们等一下。

[00:20:16] Lenny Rachitsky

English:

How long would you say it would take an engineer to build this? A typical engineer?

中文翻译:

你认为一个工程师构建这个需要多长时间？一个典型的工程师？

[00:20:22] Amjad Masad

English:

A few days, I would say to a week. I mean, if you're really good, it might be hours but it probably would take me a few days. I would say I'm like decent engineer, it'll take you a few days.

中文翻译:

我会说几天到一周。我的意思是，如果你非常厉害，可能只需要几个小时，但对我来说可能需要几天。我会说我算是个不错的工程师，这需要几天时间。

[00:20:40] Lenny Rachitsky

English:

And it took how much? 5 to 10 minutes.

中文翻译:

而它花了多久？5 到 10 分钟。

[00:20:43] Amjad Masad

English:

Yeah, and it probably cost us something in the sense.

中文翻译:

是的，而且从某种意义上说，它只花了一点点钱。

[00:20:50] Lenny Rachitsky

English:

Wow, in terms of compute.

中文翻译:

哇，指计算成本。

[00:20:52] Amjad Masad

English:

In terms of compute, yeah. Probably, I would estimate it like 15 cents or something like that.

中文翻译:

指计算成本，是的。我估计大概是 15 美分左右。

[00:20:58] Lenny Rachitsky

English:

Wow. Okay, so here it is.

中文翻译:

哇。好，它出来了。

[00:21:01] Amjad Masad

English:

Here it is. And the agent was like, "Okay, this is looking good, completed it if you want to deploy it." But I'm like, "Okay, I'm going to test it first."

中文翻译:

就在这。Agent 说：“好了，看起来不错，如果你想部署就完成了。” 但我说：“好，我要先测试一下。”

[00:21:09] Lenny Rachitsky

English:

And so currently, it's living just locally on your local host.

中文翻译:

所以目前，它只是运行在你的本地主机（Localhost）上。

[00:21:13] Amjad Masad

English:

Yeah. It's not local host, it's on a Replit but yes, it's the equivalent of local host. Because it's really easy, I can even invite you to this session. You can be here with me and so it's all online.

中文翻译:

是的。不是真正的本地主机，是在 Replit 上，但效果等同于本地主机。因为它非常方便，我甚至可以邀请你加入这个会话。你可以和我在一起，所以一切都是在线的。

[00:21:26] Lenny Rachitsky

English:

Got it.

中文翻译:

明白了。

[00:21:26] Amjad Masad

English:

So let's submit a feature. So make the product prettier. That's what a typical user might say. So we have this here, you can upvote it. I guess I can't upvote it because I'm the user that created it, but created another user. You can upvote it. But now we need to be able to move things around as the admin, so I don't know how to log into the admin panel. So I'm going to ask the agent, how do I log into the admin panel? So it might've already built the feature, and it's not exposed in the right way. It'll be able to [inaudible]

中文翻译:

那我们提交一个功能吧。比如“让产品更漂亮”。这是典型用户会说的话。我们这里有了这个请求，你可以点赞。我想我不能点赞，因为我是创建它的用户，但如果创建另一个用户，就可以点赞。但现在我们需要能以管理员身份移动东西，但我不知道怎么登录管理面板。所以我要问 Agent：我该如何登录管理面板？它可能已经构建了这个功能，只是没有以正确的方式暴露出来。它能够……（听不清）

[00:22:08] Lenny Rachitsky

English:

What I love about just watching you interact with this thing, and just real quick throughout, it feels like an engineer that is behind the scenes building this thing like on Slack and you're just talking to them. They built this thing, they're like, "Check this out, I'm done." And you're like, "Oh, okay, about how do I log into this admin panel?" And they're like, "Okay, here we go."

中文翻译:

我最喜欢看你和这东西互动的一点是，整个过程感觉就像有一个工程师在后台通过 Slack 构建这个东西，而你只是在和他聊天。他们做好了，说：“看，我做完了。”然后你说：“噢，好，但我怎么登录管理面板？”然后他们说：“好的，这就来。”

[00:22:26] Amjad Masad

English:

Yeah. So it says, "Would you like me to help you register account?" So it's creating an account, an admin account for me. So it's not only builds things, it also maintains things. In this case, it's actually doing SQL queries. It's not writing code to create an admin account for us.

中文翻译:

是的。它说：“你想让我帮你注册账户吗？”所以它正在为我创建一个账户，一个管理员账户。所以它不仅能构建东西，还能维护东西。在这种情况下，它实际上是在执行 SQL 查询，而不是编写代码来为我们创建管理员账户。

[00:22:51] Lenny Rachitsky

English:

It's insane. I want to talk about the implications of this on product development and product management and founders, but what we just witnessed is somebody, I know you do have technical abilities, but someone that didn't have to have any technical skill, build a real product that people can use in five minutes that looks good and works, and you could keep making it better by talking to this agent.

中文翻译:

这太疯狂了。我想谈谈这对于产品开发、产品管理和创始人的影响，但我们刚刚见证的是：一个人（我知道你有技术能力，但假设是一个没有任何技术技能的人）在五分钟内构建了一个人们可以使用的真实产品，它看起来不错，运行正常，而且你可以通过与这个 Agent 交谈来不断改进它。

[00:23:17] Amjad Masad

English:

I'll tell you from our experience what we're seeing, there's so many products that are empowering developers. It's a very easy calculation to say We're going to make engineers 20% better and we're going to sell it to companies and we're going to take 10% of that value, right? That's why there's so many startups now that are just trying to make engineers a little better. Our calculation is like, well, what if you made everyone developer? What does that look like? And so when we released agent and really made programming a lot easier, what we're seeing is that people, exactly like you said, people view this as a developer in their pocket essentially. What we're hearing from customers is that I'm doing things I would otherwise have to go hire a developer, but also because the activation energy is lower than going to hire developer, whether Upwork or other places, I'm building a lot more ideas that otherwise I wouldn't have built.

(00:24:26):

I think it was it called the javelin's paradox or something like that, which is when the cost of things go down, the total consumption of it goes up, which I'm not sure why they call it a paradox, but the cost of electricity goes down, maybe you would expect that the total spend goes down, but actually total spend goes up because people consume more of it. And so I think that's going to be the case of software. As the costs go down, people will just make a lot more software to improve their lives and to improve their work and start more startups and all of that.

中文翻译:

我根据我们的经验告诉你我们所看到的。现在有很多产品在赋能开发者。一个很简单的算法是：我们要让工程师的效率提高 20%，然后卖给公司，并从中抽取 10% 的价值，对吧？这就是为什么现在有这么多初创公司只是试图让工程师变得更好一点。而我们的算法是：如果你让每个人都成为开发者会怎样？那会是什么样子？所以当我们发布 Agent 并真正让编程变得容易得多时，我们看到的是，正如你所说，人们基本上把它看作是“口袋里的开发者”。我们从客户那里听到的是：我正在做一些原本必须雇开发者才能做的事；而且因为启动门槛（Activation Energy）比去 Upwork 或其他地方雇人要低，我正在实现更多原本不会去做的想法。

(00:24:26):

我想这被称为“杰文斯悖论”（Jevons Paradox）之类的，即当某种东西的成本下降时，它的总消耗量反而会上升。我不确定为什么称之为悖论，但比如电费下降了，你可能预期总支出会下降，但实际上总支出上升了，因为人们消耗了更多的电。我认为软件领域也会发生同样的情况。随着成本下降，人们会开发更多的软件来改善生活和工作，创办更多的初创公司，等等。

[00:25:05] Lenny Rachitsky

English:

So to follow that thread, what are you seeing inside of startups or even big companies in terms of how folks are already using this knowing this is the worst it will be and it will only become smarter and better these days? How are people actually using this that are say product managers or just non-technical people within startups or bigger companies?

中文翻译:

顺着这个思路，在初创公司甚至大公司内部，你看到人们是如何使用它的？考虑到现在的水平是它最差的时候，未来只会变得更聪明、更好。那些产品经理或者初创公司、大公司里的非技术人员，实际上是如何使用它的?

[00:25:26] Amjad Masad

English:

On the SMB side of things, a lot of people are building kind of back office tools. So we have real estate agents that have a lot of data, have a lot of things they want to manage in their business that building a lot of these tools, that they otherwise would have to buy, but typically when you buy, it's actually not exactly what you need. And that's kind of the problem with SaaS, like one size fits all. And so a lot of people are seeing it as sort of a SaaS replacement for in-house tools and things like that. And then when you go to the bigger companies, it's anywhere from prototyping to actually production apps to tools as well. So we've seen product managers build, like I said, like a v1 of an app and actually go out and test it with users. I can't name the company, but there's a public company that have used Replit to test a v1 of an app.

(00:26:35):

And obviously after that sort of works, they take it to the engineers and they're like, "Okay, we built this thing. We think it's a great thing. We test it with some users. Let's go actually put it on the roadmap and build it into the actual product." So you are instead of unblocking product managers from having to need engineers for everything that they want to build so they can really build the v0 or v1 of the product.

(00:27:04):

And that's super empowering for them. We saw it also with marketing departments like SpotHero has a head of marketing that actually can code decently well but use Replit to build these apps, and they built a competitive analysis application that looks at a competitor's pricing and makes sure that they're benchmarked correctly. And so it's a full stack app use database and everything and it runs on a continuous fashion. And we see sales engineers use Replit to spin up prototypes really quickly. So actually someone at X, formerly Twitter is on the partner engineering side of things, and he uses Replit agent to spin up applications and prototypes for customers to see how they can use the X API.

中文翻译:

在中小企业（SMB）方面，很多人在构建后台工具。比如我们有房地产经纪人，他们有很多数据和业务需要管理，他们正在构建很多原本需要购买的工具。但通常当你购买工具时，它并不完全符合你的需求。这就是 SaaS 的问题，即“一刀切”。所以很多人将其视为内部工具的 SaaS 替代品。在大型公司，用途从原型设计到实际的生产应用再到工具都有。我们看到产品经理构建了我所说的应用的 v1 版本，并实际拿去给用户测试。我不能透露公司名字，但有一家上市公司使用 Replit 测试了一个应用的 v1 版本。

(00:26:35):

显然，在原型运行良好后，他们会把它交给工程师，说："看，我们做了这个东西。我们觉得它很棒，已经和一些用户测试过了。让我们把它放进路线图，正式做进产品里吧。" 所以你实际上是解放了产品经理，让他们不再事事依赖工程师，他们可以真正构建产品的 v0 或 v1 版本。

(00:27:04):

这对他们来说是极大的赋能。我们也看到营销部门在使用，比如 SpotHero 的营销负责人，他其实懂一点编程，但他用 Replit 构建了这些应用，他们做了一个竞争分析应用，可以查看竞争对手的价格并确保他们的基准测试是正确的。这是一个全栈应用，使用了数据库等所有功能，并且持续运行。我们还看到销售工程师使用 Replit 快速搭建原型。实际上，X（前 Twitter）合作伙伴工程部门的一位员工就在使用 Replit Agent 为客户搭建应用和原型，展示他们如何使用 X 的 API。

[00:28:06] Lenny Rachitsky

English:

I love this. I love these examples. By the way, the demo, is there anything else you want to share about the demo before we close that out?

中文翻译:

我喜欢这些例子。顺便问一下，关于演示，在结束之前还有什么想分享的吗?

[00:28:14] Amjad Masad

English:

So it created an admin account. We can ask it with the username, password and kind of go into it and manage it but basically that's it. The app's complete in terms of what we ask for. We can send it out. I can give you a URL. Let's actually just deploy it really quickly to show people how you can deploy.

中文翻译:

它创建了一个管理员账户。我们可以问它用户名和密码，然后进去管理，但基本上就是这样。就我们的要求而言，应用已经完成了。我们可以把它发出去。我可以给你一个 URL。让我们快速部署一下，向大家展示如何部署。

[00:28:35] Lenny Rachitsky

English:

Maybe the show notes, we'll link to the app, you could check it out.

中文翻译:

也许在节目介绍里，我们会放上应用的链接，大家可以去看看。

[00:28:38] Amjad Masad

English:

Sounds good.

中文翻译:

听起来不错。

[00:28:38] Lenny Rachitsky

English:

Okay, cool. That's amazing. So this is deploying it onto some cloud provider. I don't know what you use, but...

中文翻译:

好，太棒了。所以这是把它部署到某个云服务商上。我不知道你们用的是哪家，但是……

[00:28:44] Amjad Masad

English:

We use Google Cloud. So we abstract all of that away from you, but we use Google Cloud behind the scenes.

中文翻译:

我们使用 Google Cloud。我们将所有这些都对你屏蔽了，但在后台我们使用的是 Google Cloud。

[00:28:52] Lenny Rachitsky (Ad Break)

English:

Imagine a place where you can find all your potential customers and get your message in front of them in a cost-efficient way. If you're a B2B business, that place exists, and it's called LinkedIn. LinkedIn Ads allows you to build the right relationships, drive results, and reach your customers in a respectful environment. Two of my portfolio companies, Webflow and Census, are LinkedIn success stories. Census had a 10x increase in pipeline with a LinkedIn startup team. For Webflow, after ramping up on LinkedIn in Q4, they had the highest marketing source revenue quarter to date.

(00:29:26):

With LinkedIn Ads, you'll have direct access to and can build relationships with decision makers including 950 million members, 180 million senior execs and over 10 million C-level executives. You'll be able to drive results with targeting and measurement tools built specifically for B2B. In tech, LinkedIn generated 2 to 5x higher return on ad spend than any other social media platforms. Audiences on LinkedIn have two times the buying power of the average web audience, and you'll work with a partner who respects the B2B world you operate in. Make B2B marketing everything it can be and get $100 credit on your next campaign. Just go to linkedin.com/podlenny to claim your credit. That linkedin.com/podlenny. Terms and conditions apply. Let's go down this thread actually while this is happening, just like what allows for this to be possible technology wise? What is the stack? Whatever you can share that enables this to exist.

中文翻译:

想象一个地方，你可以找到所有潜在客户，并以高性价比的方式将你的信息展示在他们面前。如果你是一家 B2B 企业，那个地方确实存在，它就是 LinkedIn。LinkedIn Ads 允许你在受尊重的环境中建立正确的关系、推动结果并触达客户。我的两家投资组合公司 Webflow 和 Census 都是 LinkedIn 的成功案例。Census 通过 LinkedIn 创业团队使销售线索增加了 10 倍。对于 Webflow，在第四季度加大 LinkedIn 投入后，他们获得了迄今为止最高的营销来源收入季度。

(00:29:26):

通过 LinkedIn Ads，你可以直接接触并与决策者建立关系，包括 9.5 亿会员、1.8 亿高级管理人员和超过 1000 万 C 级高管。你将能够使用专为 B2B 构建的定位和衡量工具来推动结果。在科技领域，LinkedIn 的广告支出回报率比任何其他社交媒体平台高出 2 到 5 倍。LinkedIn 上的受众拥有普通网络受众两倍的购买力，你将与一位尊重你所处 B2B 世界的合作伙伴合作。让 B2B 营销发挥最大潜力，并在你的下一次活动中获得 100 美元的抵用金。只需访问 linkedin.com/podlenny 即可领取。

让我们回到刚才的话题，在部署的同时，从技术角度来看，是什么让这一切成为可能？技术栈是什么？请分享一下使这一切得以存在的因素。

[00:30:26] Amjad Masad

English:

Yeah, for sure. First of all, it's all the abstractions that we built. So the way a Replit works is the very bottom layer. It's our runtime. So this is the operating system, this is the package manager, this is the language runtimes. We built a system that is able to install packages in any language, including native packages. So the AI, anytime, it needs a package. I can go here and show one of those. By the way, the AI can take screenshots as well so that it checks it works. So here you can see it is taking screenshots to make sure that the homepage is rendering. Here, you can see it wanted a drag and drop library, and so it installed that. And so it has access to all the packages across all languages, including Linux and all of that. And then the layer on top of that is the editor and the infrastructure that runs the editor, including what I described as the multiplayer editor.

(00:31:33):

And then we expose all of that infrastructure to the AI. And there's almost like a new discipline called AI Computer interfaces. So sort of like HCI is now a ACI and turns out LLMs need interfaces that are actually quite different than humans. They're trying to make them use human interfaces like Anthropic's computer use, but those are really expensive and you need to process all this images and video. So instead, for the shell for example, we give it sort of a text representation of what the shell is doing at a certain increments for package installation. We give it a certain tool for editing. We give it an editor tool that when it's writing the code, it's getting feedback on whether there are errors or not, similar to what a human sees, but it's actually old text just to make it easier. So that's AI computer interface, and obviously, all of that is sitting on foundation models. So the improvement in foundation models has allowed us to build this.

(00:32:51):

The most important model that we use is the Sonnet model from Claude, from Anthropic, and it is the best model at coding. So that's the model we use for coding, but we use models from OpenAI as well because a multi-agent system. And so we have models that are critiquing. We have manager editor model, and we have a critique model and different models will have different powers. We also train some of our models, like the embedding model for search is something we trained internally. So I actually wrote about it back in '22. I said it's going to be society of models, like products will be made of a lot of different models, and it's quite a heavy engineering project.

中文翻译:

当然可以。首先，是我们构建的所有抽象层。Replit 的运作方式最底层是我们的运行时（Runtime）。这包括操作系统、包管理器、语言运行时。我们构建了一个能够安装任何语言包（包括原生包）的系统。所以 AI 任何时候需要包时，它都能搞定。顺便说一下，AI 还可以截屏来检查运行情况。在这里你可以看到它正在截屏以确保

主页正常渲染。这里你可以看到它想要一个拖放库，于是它安装了那个库。它可以访问所有语言的所有包，包括 Linux 等等。再往上一层是编辑器和运行编辑器的基础设施，包括我提到的多人协作编辑器。

(00:31:33):

然后我们将所有这些基础设施暴露给 AI。现在几乎出现了一个新的学科，叫做 AI 计算机接口（ACI）。就像 HCI（人机交互）现在变成了 ACI，事实证明 LLM（大语言模型）需要的接口与人类截然不同。有些人试图让它们使用人类接口，比如 Anthropic 的 Computer Use，但那些非常昂贵，因为你需要处理所有的图像和视频。相反，对于 Shell（命令行），我们给它提供 Shell 在安装包时的文本表示。我们给它特定的编辑工具。我们给它一个编辑器工具，当它编写代码时，它会收到关于是否有错误的反馈，类似于人类看到的，但实际上是纯文本，为了让它更容易处理。这就是 AI 计算机接口。显然，所有这些都建立在基础模型之上。基础模型的进步让我们得以构建这一切。

(00:32:51):

我们使用的最重要的模型是来自 Anthropic 的 Claude Sonnet 模型，它是目前编程能力最强的模型。所以这是我们用于编码的模型，但我们也使用来自 OpenAI 的模型，因为这是一个多智能体（Multi-agent）系统。我们有负责批评的模型，有经理编辑器模型，有批评模型，不同的模型有不同的能力。我们还训练了一些自己的模型，比如用于搜索的嵌入（Embedding）模型就是我们内部训练的。我其实在 22 年就写过，未来将是“模型社会”，产品将由许多不同的模型组成。这是一个相当繁重的工程项目。

[00:33:47] Lenny Rachitsky

English:

To say the least. We were talking offline and you said you've been working on this since 2009 when you first built the first idea of Replit. Is that right?

中文翻译:

确实如此。我们私下聊天时你提到，你从 2009 年第一次构思 Replit 开始就在做这件事了，对吗?

[00:33:56] Amjad Masad

English:

Yes.

中文翻译:

是的。

[00:33:58] Lenny Rachitsky

English:

Oh, my God.

中文翻译:

噢，天哪。

[00:33:58] Amjad Masad

English:

Here's the deployed app. I can send it to you and you can use it and you can see my request even on the logged out page so I can register, upload it, and log in as admin and move things around. We can see what's in progress, what's completed.

中文翻译:

这是部署好的应用。我可以发给你，你可以使用它。即使在未登录页面，你也能看到我的请求。我可以注册、上传，并以管理员身份登录来移动东西。我们可以看到什么是进行中，什么是已完成。

[00:34:14] Lenny Rachitsky

English:

This looks like a product. I could see designers spending days designing, passing it to engineering, PMs, having feedback, engineers taking a few days to build it.

中文翻译:

这看起来就像一个真正的产品。我可以想象设计师花几天时间设计，交给工程部，PM 提供反馈，工程师花几天时间构建。

[00:34:26] Amjad Masad

English:

Yes.

中文翻译:

是的。

[00:34:27] Lenny Rachitsky

English:

And here's just a prompt, here's what I want.

中文翻译:

而这里只需要一个提示词：“这就是我想要的”。

[00:34:30] Amjad Masad

English:

That's right. And we can iterate on it very easily. We can also iterate on the UI. We can say, I don't like this or that, and it'll do a good job. So we can go here, we can start a new session or a new session to create an entirely new feature here, and it'll just do the right thing.

中文翻译:

没错。我们可以非常容易地进行迭代。我们也可以迭代 UI。我们可以说“我不喜欢这个或那个”，它会做得很好。我们可以去这里，开始一个新的会话来创建一个全新的功能，它就会做出正确的事情。

[00:34:47] Lenny Rachitsky

English:

And it builds from that code base. It understands here's what you've built. I want to add this thing.

中文翻译:

而且它是基于那个代码库构建的。它理解你已经构建了什么，然后在此基础上添加新东西。

[00:34:52] Amjad Masad

English:

Yes.

中文翻译:

是的。

[00:34:52] Lenny Rachitsky

English:

Okay.

中文翻译:

好的。

[00:34:53] Amjad Masad

English:

And that becomes your history, right? This was the v1 and now I'm working on this new feature, and it's almost like what engineers do in Git commit messages. By the way, it generates Git commit messages for everything that it does so you can roll back as well. And so we're trying to make it so that yes, it's for everyone, but we're trying not to abstract too much away. We want to build tools for you to learn to use. And so we want power users to be able to understand the full power of Replit and it is really deep product. I think you can spend a couple of years to kind of master it.

中文翻译:

这成了你的历史记录，对吧？这是 v1，现在我正在开发这个新功能，这几乎就像工程师在 Git 提交信息（Commit messages）中所做的那样。顺便说一下，它为它所做的每一件事都生成 Git 提交信息，所以你也可以回滚。我们努力做到：是的，它是面向所有人的，但我们尽量不把所有东西都屏蔽掉。我们想构建让你学习使用的工具。所以我们希望高级用户能够理解 Replit 的全部力量，它是一个非常深奥的产品。我认为你可能需要花几年时间才能精通它。

[00:35:40] Lenny Rachitsky

English:

I want to talk about implications, but I want to come back to something you mentioned that is incredible that people may have missed. You basically built a computer specifically designed for the AI agent to use that is a different version of a computer specifically optimized for how AI wants to use a computer.

中文翻译:

我想谈谈影响，但我先回到你提到的一个令人难以置信的点，人们可能会错过。你基本上构建了一台专门供 AI Agent 使用的计算机，它是计算机的一个不同版本，专门针对 AI 使用计算机的方式进行了优化。

[00:36:00] Amjad Masad

English:

Yeah. So there's an entire discipline called like HCI, right? It's like how do you do that-

中文翻译:

是的。有一门完整的学科叫 HCI（人机交互），对吧？就是关于你如何……

[00:36:07] Lenny Rachitsky

English:

Human-computer interaction.

中文翻译:

人机交互。

[00:36:08] Amjad Masad

English:

Yeah. So now there are papers about AI computer interfaces and interactions. And so large language models are trained on large stacks of corpus from the internet, but they're still kind of alien creatures. So they're not like humans, so they have different behaviors. It's unclear what's the best way to give it an editor. So there's so many experimentation about what's the best way to give it a view on what's editing, how many files can you show it before it starts to hallucinate. And right now, it's more of an art than science, but it's becoming more and more like a science.

中文翻译:

是的。所以现在有了关于 AI 计算机接口和交互的论文。大语言模型是在互联网的海量语料库上训练出来的，但它们仍然有点像“外星生物”。它们不像人类，所以有不同的行为。目前还不清楚给它提供编辑器的最佳方式是什么。关于如何给它展示编辑视图、在它开始产生幻觉（Hallucinate）之前能给它看多少个文件，有很多实验。目前这更像是一门艺术而非科学，但它正变得越来越像一门科学。

[00:36:55] Lenny Rachitsky

English:

This is insane. So it's a simple way to think about it. There's this foundational model, here's what I want you to build, and here's a computer to use to build it.

中文翻译:

这太疯狂了。所以一个简单的理解方式是：这里有一个基础模型，这是我想让你构建的东西，而这是一台用来构建它的计算机。

[00:37:05] Amjad Masad

English:

Yes.

中文翻译:

是的。

[00:37:05] Lenny Rachitsky

English:

How am I [inaudible]

中文翻译:

我该如何……（听不清）

[00:37:06] Amjad Masad

English:

Here's a computer with a set of tools. Here's a tool to install a package. Here's a tool to edit the code. Here's the tool to run a SQL query and also services. Here's a bunch of services you can graph from. Here's a database service, here's an object source service, here's an auth service. So you can think about it as a bunch of external services, the computer with a bunch of tools, and they're all interfacing with the foundation model.

中文翻译:

这是一台带有一套工具的计算机。这是一个安装包的工具。这是一个编辑代码的工具。这是一个运行 SQL 查询的工具，还有各种服务。这里有一堆你可以调用的服务：数据库服务、对象存储服务、身份验证服务。所以你可以把它看作是一堆外部服务、一台带有一堆工具的计算机，它们都在与基础模型进行交互。

[00:37:38] Lenny Rachitsky

English:

It's funny listening to this how, it starts to feel like the fact that we might be living in a simulation is not as far-fetched as it may feel. Like this feels like the beginnings of what a simulation computer would be.

中文翻译:

听着这些很有趣，开始觉得“我们可能生活在模拟世界中”这个说法并不像听起来那么遥不可及。这感觉就像是模拟计算机的雏形。

[00:37:50] Amjad Masad

English:

Yes. You can go really Sci-Fi on this and it's like where is it headed, right?

中文翻译:

是的。你可以从科幻的角度来看待这个问题，它会走向何方，对吧?

[00:38:03] Lenny Rachitsky

English:

Yeah.

中文翻译:

是的。

[00:38:03] Amjad Masad

English:

If we give it enough tools, let's say, I can drop it in Slack and instead of interfacing with it in this fashion, I want to interface with it in a totally autonomous way. So we actually have this feature coming up where instead of me testing it, we give it another agent. So here, instead of me interfacing with it and saying this is running or not running, we can give it another agent that is actually testing the application and then let's say interface with it entirely through Slack. And I'll say something like, give me Taylor Swift tickets the moment they land.

(00:38:49):

And so it'll build an app that continuously monitors the web for when Taylor Swift tickets land and there's an agent that's using the app to be able to get that. And you can imagine it has some kind of wallet or credit card. And then the moment it lands, it gets it. What I'm trying to say is that software, like agents being able to do software, is how AI gets more general because software runs our lives, runs the internet, runs our businesses. And so the more competent AI becomes that software, the more general they are in terms of what they can do.

中文翻译:

如果我们给它足够的工具，比方说，我可以把它放进 Slack，我不以这种方式与它交互，而是想以完全自主的方式与它交互。我们实际上即将推出一个功能，不再由我来测试，而是交给另一个 Agent。所以，不再是由我来告诉它“这行得通”或“行不通”，我们可以给它另一个 Agent 来实际测试应用程序，然后假设完全通过 Slack 与它交互。我会说类似：“泰勒・斯威夫特的门票一放出来就给我抢到。”

(00:38:49):

于是它会构建一个应用，持续监控网络上泰勒・斯威夫特门票的发布情况，并且有一个 Agent 正在使用该应用来获取门票。你可以想象它有某种钱包或信用卡。一旦放票，它就抢到了。我想说的是，软件——即 Agent 能够编写软件——是 AI 变得更通用（General）的途径，因为软件运行着我们的生活、互联网和业务。AI 在软件方面变得越能干，它们在能做的事情上就越通用。

[00:39:36] Lenny Rachitsky

English:

Okay, this can go in so many directions. I'm going to bring us back to the implications for people building products, say, product managers, founders, how does this change, that function, that skill set? What skills do you see will matter more matter less? Which functions are maybe in some danger and they should start thinking about a different career path?

中文翻译:

好，这可以引向很多方向。我想把话题拉回到对构建产品的人（比如产品经理、创始人）的影响上。这如何改变这一职能和技能组合？你认为哪些技能会变得更重要，哪些会变得不那么重要？哪些职能可能处于危险之中，他们应该开始考虑不同的职业道路?

[00:39:59] Amjad Masad

English:

One interesting persona that we're seeing is the CEO, the CEO of startup. Andrew Wilkinson from Tiny is a big user. And so these people are typically creatives. They built a company, they hired people. A lot of them can't code. A lot of them are designers or product managers or something else. And you can imagine a bottleneck, you can imagine a bunch of ideas in their head, and the ideas have to translate through them talking. And then someone else listening to them and assuming that someone else actually understands what they say, and then that's someone else going and trying to build what they want to build. And also assuming that person has time, because a lot of times, your engineers are kind of stuck building the current thing. They're not thinking about the future thing. And so what gets me excited is a lot of these CEOs are building the future concept, the next product they're going to build, the next, say, company they're going to build.

(00:41:08):

And so it unlocks the creativity and again, sort of unblocks them from that. And look, it's a v1 of the product but it can push things forward. You can touch it, you can feel it, you can say, okay, this really has lags and we should work on it. You give it to your engineers and they can improve on it from there. So that's one persona but I'm really excited about it, the CEO/founder. In companies, one of the things that I think is sort of hard about tech companies is these silos between designers, product managers, and engineers. And everyone feels that pain of, we have low bandwidth communication, which is language and then text on Slack and Zoom calls. And it leads to a lot of frustration, because it's really easy to misinterpret people and again, leads to siloing where people working on something, and then you pass it on to the next team and it's not really what they expect.

(00:42:30):

That happens a lot between designers and engineers, but the common language that everyone shares is code. Ultimately in software tech companies, everything that we're talking about need to eventually flush out in terms of code. And so what if the language becomes actually working prototypes and working applications? For example, we have a Figma extension that translate Figma mocks into React that runs on Replit. So instead of giving the engineers just mocks or screenshots, whatever, you just say, oh, here's a bunch of React code, just make sure it runs on our infrastructure but don't mess with it, don't move the pixels around. And so I think it just opens up silos of the companies, make communication around product a lot more concrete, because I can give you a working prototype and that'll change how people work, if you can imagine that everyone can make software. It's really kind of a radical reimagining of not just what tech companies are but really what most companies are because everyone can be more general.

中文翻译:

我们看到的一个有趣的画像是 CEO，初创公司的 CEO。Tiny 的 Andrew Wilkinson 就是一个重度用户。这些人通常是创意人士。他们创办了公司，雇佣了员工。他们中很多人不会编程，很多人是设计师、产品经理或其他出身。你可以想象一个瓶颈：他们脑子里有一堆想法，这些想法必须通过他们的交谈来转化。然后别人听他们说，并假设那个人真的理解了他们的意思，然后那个人再去尝试构建他们想要的东西。还要假设那个人有时间，因为很多时候，你的工程师正忙于构建现有的东西，他们没在考虑未来的东西。所以让我兴奋的是，很多 CEO 正在构建未来的概念，他们要做的下一个产品，或者下一家公司。

(00:41:08):

所以它释放了创造力，再次打破了瓶颈。看，虽然只是产品的 v1 版本，但它可以推动事情向前发展。你可以触摸它，感受它，你可以说："好吧，这确实有延迟，我们应该改进它。" 然后你把它交给工程师，让他们在此基础上进行改进。所以这是我非常兴奋的一个画像：CEO/创始人。在公司里，我认为科技公司比较困难的一点是设计师、产品经理和工程师之间的孤岛。每个人都能感受到那种痛苦：我们的沟通带宽很低，也就是语言，然后是 Slack 上的文字和 Zoom 通话。这导致了很多挫败感，因为人们很容易误解对方，再次导致孤岛化——人们在做某件事，然后把它传给下一个团队，结果却不是他们预期的那样。

(00:42:30):

这种情况在设计师和工程师之间经常发生，但每个人共享的通用语言是代码。最终在软件科技公司，我们讨论的所有内容最终都需要以代码的形式呈现。那么，如果沟通语言变成了实际运行的原型和应用程序会怎样？例如，我们有一个 Figma 插件，可以将 Figma 原型转换为在 Replit 上运行的 React 代码。所以，你不再只是给工程师原型图或截图，你直接说："噢，这是一堆 React 代码，只要确保它在我们的基础设施上运行就行，但别乱动它，别移动像素。" 所以我认为这打破了公司的孤岛，让围绕产品的沟通变得更加具体，因为我可以给你一个可以运行的原型。如果你能想象每个人都能制作软件，那将改变人们的工作方式。这不仅是对科技公司的激进重构，也是对大多数公司的激进重构，因为每个人都可以变得更加通用。

[00:44:04] Lenny Rachitsky

English:

So say you're a PM listening to this, an engineer or designer, what skills do you think if you were one of these folks? If you were in building Replit right now, what kind of skills would you suggest folks focus on more and would you think are just like, okay, this can be less valuable in the future, don't worry about these sorts of things? And you can either pick one of those three functions or all three.

中文翻译:

假设你是一个正在听节目的 PM、工程师或设计师，如果你是其中一员，如果你现在正在构建 Replit，你会建议大家多关注哪些技能？你认为哪些技能在未来会变得不那么有价值，不用再担心了？你可以从这三个职能中选一个，或者三个都谈谈。

[00:44:25] Amjad Masad

English:

I think a very important scale that's perhaps harder to develop but it's worth working on is being generative, being more generative, being able to generate new ideas quickly, because you can think about it as a factory line. So you have ideas, you have the production of these ideas or the initial production of these ideas, and then you have other people that want to consume these ideas or work with you on these ideas. And so typically you're bottlenecked by the middle part where your ideas are kind of like they're a lot of them and they're not fitting in, because they need to be made and they need to be made quickly. And so now you open up that bottleneck. So now actually making things is a lot easier. Actually, you become limited by how fast you can generate ideas, and I find that true of myself as well. I consider myself quite generative, but now I have this tool and I can build a lot more and explore a lot more and I'm finding that, well, actually, I'm running out of ideas sometimes.

(00:45:53):

So training that muscle I think is a good thing. I think learning a little bit of coding and not the traditional way of learning coding. If you go to a coding bootcamp, they're going to start with what is Git? Actually my co-founder, Haya, was a designer. When we're first building the Replit together, she went to WebAssembly to do a coding course. And the first day, they spent this whole time on Git and she's like, "What is that? What does it do?" I still don't know what it exactly does, but it's like you're inverting the process, you're giving the tool before the actual problem. And so I think all of that stuff, you don't have to worry about, so things that you don't have to worry about. I think a lot of the, as a PM, as a designer, as someone who's not in your code editor every day, don't worry about all the tooling.

(00:47:09):

And if you learn a little bit of coding just by talking to an AI, doing a little bit of debugging, building something with Replit, running into a problem and trying to fix it just using AI, you'll learn a bit of coding. And I have this that's been called not by me, dubbed as Amjad's Law, which is the return on investment for learn to code is doubling every six months. And really just learning a little bit of that skill, learning a bit of skill about how to prompt AI, how to read code, and be able to debug it. Every six months, that's netting you more and more power because you're going to be able to create a lot more. It's going to be easier to create. You're going to be able to create a lot more complete things. So that's another skill that I think could be necessary.

中文翻译:

我认为一个非常重要、可能更难培养但值得努力的技能是“生成能力”（Being generative），即能够快速产生新想法。你可以把它想象成一条工厂流水线：你有想法，你有这些想法的生产（或初始生产），然后有其他人想要消费这些想法或与你合作。通常情况下，你的瓶颈在于中间部分——你的想法很多，但无法落地，因为它们需要被制造出来，而且需要快速制造。现在你打破了这个瓶颈。所以现在实际动手做东西变得容易得多。事实上，你的限制变成了你产生想法的速度。我发现我自己也是这样。我认为自己很有生成能力，但现在有了这个工具，我可以构建更多、探索更多，我发现，哎呀，有时候我竟然没主意了。

(00:45:53):

所以锻炼这块“肌肉”我认为是一件好事。我认为要学习一点编程，但不是传统的学习方式。如果你去参加编程训练营，他们会从“什么是 Git”开始。实际上我的联合创始人 Haya 以前是设计师。当我们第一次一起构建 Replit 时，她去参加了一个 WebAssembly 编程课程。第一天，他们把所有时间都花在 Git 上，她就问：“那是什么？它是干什么的？” 我现在也说不准它到底是干什么的，但这就像是你把过程颠倒了，在遇到实际问题之前先给了工具。所以我觉得所有那些东西，你都不必担心。作为 PM、设计师，或者不是每天都盯着代码编辑器的人，不要担心所有的工具链。

(00:47:09):

如果你只是通过与 AI 交谈、做一点调试、用 Replit 构建一些东西、遇到问题并尝试仅使用 AI 来修复它，你就会学到一点编程。我有一个被别人（不是我）称为“Amjad 定律”的说法，即学习编程的投资回报率每六个月翻一番。真的，只要学习一点点这项技能，学习一点关于如何提示 AI、如何阅读代码以及如何调试的技能。每六个月，这都会为你带来更多的力量，因为你将能够创造更多。创造将变得更容易。你将能够创造出更完整的东西。所以我认为这是另一项必要的技能。

[00:48:15] Lenny Rachitsky

English:

This is super interesting. Okay, so this last point, you made Amjad's Law. It's interesting because when people, as someone's listening to this, I could see them being like, engineers are in trouble. Why do you need engineers at this point? These agents are building the code. Your point is specific engineering skills are going to be incredibly valuable and more and more. How often are they doubling would you say? Every year you said?

中文翻译:

这非常有趣。好，关于最后一点，你提出的“Amjad 定律”。很有意思，因为当人们听这个节目时，我能预见到他们会想：工程师有麻烦了。既然这些 Agent 都在写代码，为什么还需要工程师？而你的观点是，特定的工程技能将变得极其有价值，而且价值越来越高。你说它们翻倍的频率是多少？你说是每年？

[00:48:40] Amjad Masad

English:

No, every six months.

中文翻译:

不，是每六个月。

[00:48:41] Lenny Rachitsky

English:

Every six months, these specific engineering skills are becoming more valuable. And the idea is you don't need to know everything. You don't need to know the foundation, to build the app as much. It's more to unblock the agent and understand the mental model of how this stuff is built so that you can move forward fast.

中文翻译:

每六个月，这些特定的工程技能就会变得更有价值。核心理念是你不需要知道所有事情。你不需要像以前那样了解构建应用的底层基础。更多的是为了给 Agent “疏通”瓶颈，理解这些东西是如何构建的思想模型，以便你能快速前进。

[00:48:59] Amjad Masad

English:

That's right. That's right. Understanding the basic components of it, I would say, yes.

中文翻译:

没错。没错。我会说，是理解它的基本组成部分。

[00:49:03] Lenny Rachitsky

English:

Yeah, so it's like we need new engineering schools to teach you these very specific skills versus spending years on algorithms.

中文翻译:

是的，所以就像我们需要新的工程学校来教你这些非常具体的技能，而不是花几年时间研究算法。

[00:49:11] Amjad Masad

English:

And I think no one has done that yet, and I think this is a big business probably ready to get built. It's like AI native coding. It's totally different than traditional coding. That's why on Hacker News, there's so much skepticism about AI native coding tools, because they're like yeah, it's a glorified autocomplete. And I understand if you're writing operating system, kernels, it's not really doing that much for you, but if you're building products, it's building it for you at this point. And so if you're starting a school to teach AI native coding, you would skip so much of computer science and the basic tools, and you would teach the basic idea of how to structure an app, and then you would teach prompting and then you would teach, I think a little bit of debugging. I think debugging is quite a good skill right now to learn.

中文翻译:

我认为目前还没有人做这件事，我认为这是一个可能即将爆发的大生意。这就像是“AI 原生编程”。它与传统编程完全不同。这就是为什么在 Hacker News 上，人们对 AI 原生编程工具持怀疑态度，因为他们觉得“这不过是高级版的自动补全”。我理解如果你是在编写操作系统内核，它确实帮不上什么忙，但如果你是在构建产品，它现在就是在为你构建。所以如果你创办一所教授 AI 原生编程的学校，你会跳过大量的计算机科学基础和基本工具，你会教授如何构建应用的基本思路，然后教授提示词工程，接着教授——我认为是一点调试技巧。我认为调试在目前是一项非常好的技能。

[00:50:22] Lenny Rachitsky

English:

Interestingly, if you want to be good at debugging, there's a lot you need to understand, which is basically what you're saying is that's the subset of things to understand is things that break. And to do that, you have to understand how it all works. What are servers? What are APIs? All these things. Okay, so we've been talking about how this is very good right now, building a prototype, building a v1, MVP, people can use it, you can deploy it. You deploy this app, people can start using it, and there's a scale it can reach. Do you see a future where you can build a Salesforce sized business fully Replit or other tools that can scale to hundreds of billions of dollars of value? Or is there just going to always be some limit of like, you need actual engineers and designers sitting on this thing, building it, thinking it? Awesome.

中文翻译:

有趣的是，如果你想擅长调试，你需要理解很多东西，这基本上就是你所说的——需要理解的子集就是“会出故障的东西”。而要做到这一点，你必须理解它是如何运作的。什么是服务器？什么是 API？所有这些。好，我们一直在谈论它现在非常出色，构建原型、构建 v1、MVP，人们可以使用它，你可以部署它。你部署了这个应用，人们可以开始使用，它能达到一定的规模。你是否预见到未来可以完全通过 Replit 或其他工具构建一个 Salesforce 规模的企业，并扩展到数千亿美元的价值？还是说总会有一个限制，比如你需要真正的工程师和设计师坐在那里构建它、思考它？

[00:51:06] Amjad Masad

English:

If my law is directionally correct, even if the months are not, I'm not exactly right that the duration is correct, you're going to see a compounding effect of the power. It's actually quite hard to convince yourself. But if you really convince yourself that we are on a massive scale of improvement in AI, then the answer is yes. And it's absurd to my engineering mind that I'm saying it is, but know Ray Kurzweil, this futurist talks about how exponentials are really hard for humans to grasp. And so actually when we started building the agent, I told the team, it's easy and we fall in this trap before. It's easy to build and optimize for today. In '22, we built Copilot-like thing and autocomplete. We train our own models, we optimize the hell out of them. But at some point, that modality was kind of not the right modality, which is the autocomplete modality.

(00:52:11):

And the right modality is actually this, I think for now, as being able to chat inside the programming environment and for the agent to create things for you. But in order for us to make that bat, a year ago the models were actually not there. The models could not do this, but we were like, okay, we're going to build for the models that are landing in six months. And truly six months later, the model started to land that are capable of this, of the reasoning that we needed and whatever. And so that was like saw it if you want, which is, oh, wow, we switched to it and the reasoning improved so much. And six months later, you have a son of you too. And so it's really almost like a six months cadence. And so if we're really on this trajectory, then I would say next year, you're able to just scale and maybe you get thousands of users paying you.

(00:53:08):

The AI can do maintenance. We already showed the AI doing SQL queries and doing migrations, so I will be able to do maintenance, debugging, things like that. I think where it gets really tough is that when you're hitting scale and you want to architect a system that is resilient, and so that means you would start sharding databases and you would start using different queue systems and components and things like that. And I think the AI needs to have access to the entire suite of tools to be able to do this.

(00:53:49):

And I think that's going to be the next bottleneck. And I think the AI needs to be a lot more reliable at doing that. But I could imagine whatever, five years from now, someone running a billion dollar company with zero employees where it's like the support is handled by AI, the development is handled by AI, and you're just building and creating this thing that people are finding valuable and are paying you for it. That being said, it's worth thinking about the economics of it. If the cost of software goes down a lot, then what is the price that you can charge on software? So can you actually build the next Salesforce if anyone can generate Salesforce? And then the question is, and this is why I emphasize being generative, because I think then the thing that will make you better is by being able to iterate and improve the thing really quickly and generate new ideas.

中文翻译:

如果我的定律在方向上是正确的（即使月份不一定准确），你会看到能力的复利效应。这其实很难说服自己。但如果你真的相信我们正处于 AI 的大规模进步中，那么答案是肯定的。作为一个有工程思维的人，我说出这个答案也觉得很荒谬，但你知道未来学家 Ray Kurzweil 谈到过，人类很难理解指数级增长。所以实际上当我们开始构建 Agent 时，我告诉团队：为今天而构建和优化很容易，我们以前也掉进过这个陷阱。22 年我们做了类似 Copilot 的东西和自动补全，我们训练自己的模型，拼命优化。但到某个点，那种模式（自动补全模式）就不再是正确的模式了。

(00:52:11):

我认为目前的正确模式是这种：能够在编程环境中聊天，并让 Agent 为你创造东西。但为了做这个赌注，一年前模型其实还达不到要求。模型做不到这些，但我们当时想：好吧，我们要为六个月后出现的模型而构建。果然六个月后，具备这种推理能力的模型开始出现了。所以就像 Sonnet 出现时，我们切换过去，推理能力提升了

巨大。再过六个月，又有了新版本。所以这几乎是一个六个月的节奏。如果我们真的处于这条轨迹上，那么我会说明年，你就能实现规模化，也许会有成千上万的用户付钱给你。

(00:53:08):

AI 可以进行维护。我们已经展示了 AI 执行 SQL 查询和迁移，所以它将能够进行维护、调试等工作。我认为真正困难的地方在于，当你达到一定规模并想要架构一个具有弹性的系统时，这意味着你需要开始对数据库进行分片（Sharding），开始使用不同的队列系统和组件等等。我认为 AI 需要访问整套工具才能做到这一点。

(00:53:49):

我认为这将是下一个瓶颈。AI 在做这些事情时需要变得更加可靠。但我可以想象，五年后的某一天，有人经营着一家十亿美元的公司，却没有任何员工，支持由 AI 处理，开发由 AI 处理，而你只是在构建和创造这个人们认为有价值并愿意为此付费的东西。话虽如此，软件的经济学也值得思考。如果软件成本大幅下降，那么你能收多少钱？如果任何人都能生成一个 Salesforce，你还能建立下一个 Salesforce 吗？所以这就是为什么我强调“生成能力”，因为到那时，让你脱颖而出的将是快速迭代、改进并产生新想法的能力。

[00:55:01] Lenny Rachitsky

English:

And stay ahead of all the other people building these tools so quickly. Oh, my God. An interesting other kind of mental model I'm seeing as you talk about this sort of thing is not to offend religious folks, but there's this concept of God of the gaps. I imagine you've heard that where it's like God explains all the things that we don't yet understand. And over time that kind of space shrinks and God's like all the things we don't get yet, those gaps. That was God that proves there needs to be a God. And it feels like right now, humans are the gaps in these tools or these agents you talk about that you can hire within Replit are fixing these little gaps. And over time, AI will fix these things themselves.

中文翻译:

并且要领先于所有其他快速构建这些工具的人。噢，天哪。听你谈论这些，我想到另一个有趣的思维模型（无意冒犯宗教人士），就是“缝隙中的上帝”（God of the gaps）这个概念。我想你可能听说过，即上帝解释了所有我们尚不理解的事物。随着时间的推移，那个空间会缩小，上帝就成了“我们还不理解的所有事物”，即那些缝隙。这证明了上帝的存在。感觉现在，人类就是这些工具中的“缝隙”，或者你提到的在 Replit 中可以雇佣的 Agent 正在修复这些小缝隙。随着时间的推移，AI 会自己修复这些东西。

[00:55:44] Amjad Masad

English:

That's right.

中文翻译:

没错。

[00:55:44] Lenny Rachitsky

English:

And these gaps will shrink.

中文翻译:

而这些缝隙会缩小。

[00:55:47] Amjad Masad

English:

Unless we hit some fundamental limit and the current regime of AI, which I'm not an expert about how far transformers could scale, but I feel like we found the thing that could scale pretty far, but maybe there are limitations in data or other things like that that we could be surprised by. But if there isn't, then we are on a massive trajectory of removing these gaps quickly.

中文翻译:

除非我们遇到了某种根本性的限制，或者当前的 AI 范式（我不是 Transformer 能扩展到多远的专家）遇到了瓶颈。但我觉得我们已经找到了可以扩展到很远的东西，不过也许在数据或其他方面存在一些会让我们感到意外的限制。但如果没有，那么我们正处于快速消除这些缝隙的宏大轨迹上。

[00:56:23] Lenny Rachitsky

English:

Yeah, very true. We have no idea. We keep thinking it's just going to keep going, but maybe it'll stop at some point. I could keep going and going, but I think we should also let people go play with these things and process all the things we've been talking about. Is there anything else that you think might be helpful for folks to think about or learn or study?

中文翻译:

是的，非常正确。我们无从知晓。我们一直以为它会持续发展，但也许在某个点会停下来。我可以一直聊下去，但我想我们也该让大家去亲自尝试这些东西，并消化我们讨论的内容。你认为还有什么对大家思考、学习或研究有帮助的吗？

[00:56:43] Amjad Masad

English:

I'll give advice to founders or leaders at companies. The way we work is going to change rapidly, and it's important to be resilient to that change. One thing that I think is really difficult now is having roadmaps, especially if you're doing anything in AI, but really anything that AI could affect, you want to be able to react to it really quickly. And so when the Anthropic drop the computer use set of capability, we slaughtered our roadmap because we don't really have an explicit roadmap. We immediately jumped on it and started building things and we launched some things around it. We're going to be doing more with it, but there's going to be capabilities that are going to drop. And you want to really, in some cases, if it really affects your business, you want to be able to jump on it really, really quickly. So being agile, not being stuck with roadmaps, being able to just say, oh, we're just going to switch priorities right away, is going to be super important.

(00:57:59):

Not being, like I said, with silos at Replit, there's so many people that are on the scale of designer to engineer, designer, product manager. Actually, I mentioned Amman earlier. He started as a designer at Replit, and now as a product manager. We have people who start as designers, become engineers, and we have people in the middle and we're comfortable with that design engineers that fit at different parts of the scale. And the design engineers go to the design correct meetings and some designers go to the engineering meetings. You got to be fluid because again, when designers can code and engineers can design, I mean it really becomes, you can't have a lot of structure around that. So you want to build a culture and you want to build an environment or milieu that is really, really flexible, which is uncomfortable for a lot of people.

中文翻译:

我会给公司的创始人或领导者一些建议。我们的工作方式将迅速改变，对这种改变保持韧性非常重要。我认为现在非常困难的一件事是制定路线图（Roadmap），特别是如果你在做任何与 AI 相关的事情，或者任何可能受 AI 影响的事情，你必须能够迅速做出反应。所以当 Anthropic 发布 Computer Use 功能时，我们“砍掉”了我们的路线图，因为我们其实没有明确的路线图。我们立即投入其中并开始构建，并围绕它发布了一些东西。我们还会做更多，但未来还会有更多能力发布。在某些情况下，如果它真的影响了你的业务，你需要能够非常、非常迅速地跟进。所以保持敏捷，不被路线图束缚，能够随时说“我们要立即切换优先级”，这将变得至关重要。

(00:57:59):

正如我所说，不要有孤岛。在 Replit，有很多人处于设计师到工程师、设计师到产品经理的光谱上。实际上，我之前提到的 Amman，他在 Replit 是从设计师开始的，现在是产品经理。我们有从设计师开始变成工程师的人，也有处于中间地带的人，我们对这种“设计工程师”感到很适应，他们分布在光谱的不同位置。设计工程师会参加设计评审会议，一些设计师也会参加工程会议。你必须保持流动性，因为当设计师能编程、工程师能设计时，你就不能有太多的条条框框。所以你要建立一种文化，建立一个非常、非常灵活的环境或氛围，虽然这对很多人来说是不舒服的。

[00:59:03] Lenny Rachitsky

English:

Man, the future is wild. Everyone's a hybrid person now. Let me just actually double down on what you just said, which I think is really interesting. It's almost like if you're an engineer, where your skill set will become most valuable is unblocking these AI tools and knowing debugging and figuring out how to allow it to go further and further and further. Within PM and design land, based on what you're describing, where the skills will become more valuable is generating ideas, almost like finding opportunities, discovery, finding what problems need to be solved, and then articulating that as clearly as possible to the AI tooling.

中文翻译:

伙计，未来太疯狂了。现在每个人都是“混合型”人才。让我再强调一下你刚才说的话，我觉得非常有趣。这几乎就像：如果你是工程师，你最有价值的技能将是为这些 AI 工具疏通瓶颈、精通调试，并搞清楚如何让它走得越来越远。而在 PM 和设计领域，根据你的描述，最有价值的技能将是产生想法，几乎就像寻找机会、探索、发现需要解决的问题，然后尽可能清晰地将其表达给 AI 工具。

[00:59:44] Amjad Masad

English:

That's right.

中文翻译:

没错。

[00:59:45] Lenny Rachitsky

English:

Super interesting.

中文翻译:

非常有趣。

[00:59:45] Amjad Masad

English:

Yeah, this is a very crisp sort of advice that people can follow today, I think.

中文翻译:

是的，我认为这是人们今天就可以遵循的非常清晰的建议。

[00:59:50] Lenny Rachitsky

English:

Oh man, what a world. Okay, I'm jot. This is incredible. My mind is racing. I've got to go build some apps immediately.

中文翻译:

噢伙计，这是个什么样的世界啊。好，Amjad，这太不可思议了。我的大脑在飞速运转。我得立刻去构建一些应用。

[00:59:58] Amjad Masad

English:

Yes, you've back. Love that.

中文翻译:

是的，欢迎回来。我喜欢这个。

[01:00:00] Lenny Rachitsky

English:

I will do that. So just to leave listeners with a couple things. One is just, what should they know? Where do they find you? How do they try Replit? Anything else other than just go to replit.com?

中文翻译:

我会去做的。在结束前给听众留几件事。一是，他们应该知道什么？在哪里可以找到你？如何尝试 Replit？除了访问 replit.com 之外还有别的吗？

[01:00:10] Amjad Masad

English:

Yeah, just go to replit.com. It's an open beta right now. We're kind of quickly improving and going to exit beta I think in a few weeks. But if you're comfortable testing something that's not perfect, go to replit.com. If you subscribe to our core plan, you should be able to access the agent and start using it. And we are, I think the place where we're most active is Twitter. So Twitter are like X, the handle Replit, R-E-P-L-I-T or my handle @amasad.

中文翻译:

是的，直接去 replit.com 即可。目前处于公开测试阶段。我们正在快速改进，我想几周后就会结束公测。但如果你愿意测试一些还不完美的东西，就去 replit.com 吧。如果你订阅了我们的核心计划（Core Plan），你应该就能访问 Agent 并开始使用。我们最活跃的地方应该是 Twitter，也就是 X，账号是 Replit 或者我的账号 @amasad。

[01:00:45] Lenny Rachitsky

English:

Oh, yeah. One other thing I wanted to make sure we had a chance to touch on is you're working on something new, something that's coming in the very new future, maybe the day this episode drops. Talk about that.

中文翻译:

噢对了。我还想确保我们有机会提到一件事：你正在开发一些新东西，很快就会面世，也许就在这期节目发布的那天。谈谈那个吧。

[01:00:56] Amjad Masad

English:

All right. So depending on when the episode is coming out, this could be the first time people hear about it. But we have this product called agent. It is sort of high agency, does everything from setting up the project and all of that. And so now, we are working on assistant. Assistant is let's say the cousin of agent. It is a little less powerful but a lot more controllable. You can focus on features or areas of the code that you want to change and you still don't have to know how to code, but it is a lot more manageable and it is a lot faster.

(01:01:38):

So you saw how it took some time to kind of create the project and code some of the things. Assistant is in the order of milliseconds and seconds to be able to respond to you. And so again, as I talk about the idea of tools, we want people to have as much power and autonomy as possible. And so there are certain instances where agent is the best. It's going to do the debugging for you, it's going to create the database for you. But if you want more control, assistant is going to give you that.

中文翻译:

好的。取决于这期节目什么时候播出，这可能是人们第一次听说它。我们有一个叫 Agent 的产品，它具有很高的自主性，能完成从建立项目到所有的一切。现在，我们正在开发 Assistant（助手）。Assistant 可以说是 Agent 的表亲。它的功能稍微弱一点，但可控性强得多。你可以专注于想要更改的功能或代码区域，你仍然不需要懂编程，但它更容易管理，而且速度快得多。

(01:01:38):

你刚才看到 Agent 创建项目和编写代码花了一些时间。Assistant 的响应速度在毫秒和秒的级别。所以，再次回到工具的理念，我们希望人们拥有尽可能多的权力和自主权。在某些情况下，Agent 是最好的，它会为你调试，为你创建数据库。但如果你想要更多的控制权，Assistant 会满足你。

[01:02:09] Lenny Rachitsky

English:

Just so folks totally understand what this is going to do for them. What's the mental model for what this is? If it's like a person, we're helping you out.

中文翻译:

为了让大家完全理解这能为他们做什么。它的思想模型是什么？如果把它比作一个人，它是如何提供帮助的？

[01:02:17] Amjad Masad

English:

Agent is like having a developer work. You give them the PRD, right? And they're going to go and build the thing. Assistant is like you're sitting next to them. So they built the thing and now you walk over to their desk and you say, let me move this button. Three pixels to the left. Let me change this thing. So small increments of changes that you want happen really quickly and you want it reliably, that will give you that. So it's just much faster iteration on UI and things like that.

中文翻译:

Agent 就像是雇了一个开发者。你给他们 PRD，对吧？然后他们就去把东西做出来。Assistant 就像是你坐在他们旁边。他们做好了东西，现在你走到他们的办公桌前说：“让我把这个按钮向左移动三个像素。让我改一下这个东西。” 那些你想要快速且可靠地实现的微小增量变化，它都能帮你实现。所以它在 UI 迭代等方面要快得多。

[01:02:58] Lenny Rachitsky

English:

Incredible. The future is wild. Final question I always ask everybody, how can listeners be useful to you?

中文翻译:

不可思议。未来真是疯狂。最后一个我总是问每个人的问题：听众可以为你提供什么帮助？

[01:03:04] Amjad Masad

English:

Come work at Replit. We have a PM role. I think up if you're product manager. We're hiring engineers and product managers. So come work at Replit or refer someone to Replit, especially if you're like our tools and you want them to get better. The best way to do that is to get us great people we can hire.

中文翻译:

来 Replit 工作吧。我们有一个 PM 职位。如果你是产品经理，可以关注一下。我们正在招聘工程师和产品经理。所以，来 Replit 工作，或者向 Replit 推荐人才，特别是如果你喜欢我们的工具并希望它们变得更好。最好

的方式就是为我们引荐我们可以雇佣的优秀人才。

[01:03:25] Lenny Rachitsky

English:

Well, you're about to get a flood of product managers applying. Goodluck.

中文翻译:

好吧，你马上就会收到一大波产品经理的申请。祝你好运。

[01:03:28] Amjad Masad

English:

Amazing. I love that.

中文翻译:

太棒了，我喜欢。

[01:03:30] Lenny Rachitsky

English:

Amjad, thank you so much for being here. This was incredible.

中文翻译:

Amjad，非常感谢你能来。这太精彩了。

[01:03:33] Amjad Masad

English:

Thank you. Thank you for your podcast and the community that you've built and newsletter and everything. It's been awesome to watch.

中文翻译:

谢谢。谢谢你的播客，以及你建立的社区、时事通讯和所有的一切。看到这些真的很棒。

[01:03:40] Lenny Rachitsky

English:

Thanks, man. Appreciate that. 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:

谢谢，伙计。非常感谢。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上订阅本节目。此外，请考虑给我们评分或留下评论，因为这确实能帮

助其他听众找到这个播客。你可以在 lennyspodcast.com 找到所有往期节目或了解更多关于本节目的信息。下期节目再见。