MELISSA PERRI

LENNY'S PODCAST

BILINGUAL TRANSCRIPT

ORIGINAL BY

Lenny Rachitsky

@lennysan • x.com/lennysan

ANALYSIS BY

@Penny777 • x.com/penny777

Generated: 2026-01-16

Melissa Perri - 双语对照

Lenny's Podcast: Melissa Perri - Product Owners, Scrum, and SAFe

[00:00:00] Lenny Rachitsky

English:

There's this whole concept of SAFe, basically Scaled Agile, right?

中文翻译:

这里有一个关于 SAFe 的完整概念，基本上就是大规模敏捷（Scaled Agile），对吧?

[00:00:03] Melissa Perri

English:

Scaled Agile Framework came out of the desire to figure out how do we scale Scrum and different processes. I do not recommend using SAFe. Every single person I have talked to who likes SAFe, found success with SAFe, they ended up ripping it up and making it into something else.

中文翻译:

大规模敏捷框架（SAFe）源于人们想要弄清楚如何扩展 Scrum 和不同的流程。我不建议使用 SAFe。我交谈过的每一个喜欢 SAFe 并在其中获得成功的人，最终都把它拆解并改造成了别的东西。

[00:00:18] Lenny Rachitsky

English:

You've been up close and personal with a lot of companies working with product owners, Scaled Agile, and all these things.

中文翻译:

你曾近距离接触过许多使用产品负责人（Product Owner）、大规模敏捷以及这类东西的公司。

[00:00:23] Melissa Perri

English:

This product owner role did not emerge from product management as we know it today. It was a way to help the developers prioritize what to work on. I ended up going to a ton of Agile conferences and speaking about product management, and I started to learn that there was this product owner role in Scrum.

中文翻译:

这个“产品负责人”角色并不是从我们今天所熟知的产品管理中演变而来的。它最初是一种帮助开发人员确定工作优先级的方法。后来我参加了大量的敏捷会议并发表关于产品管理的演讲，我才开始了解到 Scrum 中有这样一个产品负责人的角色。

[00:00:39] Lenny Rachitsky

English:

It feels like it's growing. More and more companies are adopting this as the way to work.

中文翻译:

感觉它正在增长。越来越多的公司正在采用这种方式作为工作模式。

[00:00:44] Melissa Perri

English:

A lot of large companies turn to Scrum or to the frameworks, and it's because they traditionally didn't grow up building software. When you look at agile methodologies, what we're really saying there is we want to be able to move quickly and deliver great value to customers. If you embrace those principles, you're going to do well.

中文翻译:

许多大公司转向 Scrum 或这些框架，是因为它们传统上并不是靠开发软件起家的。当你审视敏捷方法论时，我们真正的意思是希望能够快速行动并为客户提供巨大的价值。如果你拥抱这些原则，你就会做得很好。

[00:01:05] Lenny Rachitsky

English:

Today my guest is Melissa Perri. Melissa is a legend in the product management community. She's the author of the foundational book Escaping the Build Trap, and her most recent book Product Operations. She's also the CEO and founder of The Product Institute, which trains product managers at all levels. She's trained PMs at almost every Fortune 500 company at this point, and in our conversation we dive deep into a topic that I don't spend a lot of time on on this podcast, product owners, Scrum, Scaled Agile, and building product at very large non-tech companies.

(00:01:38):

Melissa shares the history behind these ways of working, what she's seen work and not work when companies roll out these frameworks, and most importantly what you can do as a leader at one of these companies and as a product owner working in one of these companies to level up your organization and yourself. I learned a ton from this conversation and I'm really curious to hear what you think since we don't cover this kind of stuff on this podcast too much. 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 future episodes, and it helps the podcast tremendously. With that, I bring you Melissa Perri. Melissa, thank you so much for being here and welcome to the podcast.

中文翻译:

今天的嘉宾是 Melissa Perri。Melissa 是产品管理界的传奇人物。她是基础著作《逃离构建陷阱》（Escaping the Build Trap）以及最新著作《产品运营》（Product Operations）的作者。她还是 The Product Institute 的首席执行官兼创始人，该机构负责培训各级产品经理。到目前为止，她几乎培训过所有财富 500 强公司的 PM。在我们的对话中，我们将深入探讨一个我在本播客中不常涉及的话题：产品负责人、Scrum、大规模敏捷，以及在大型非科技公司中构建产品。

(00:01:38):

Melissa 分享了这些工作方式背后的历史，她在公司推行这些框架时看到的成功与失败案例，最重要的是，作为这些公司的领导者或在其中工作的产品负责人，你可以做些什么来提升组织和你自己。我从这次对话中学到了很多，也很想听听你们的想法，因为我们平时不怎么聊这类内容。如果你喜欢这个播客，别忘了在订阅并关注。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 Melissa Perri。Melissa，非常感谢你能来，欢迎来到播客。

[00:02:19] Melissa Perri

English:

Thanks, Lenny. Thanks for having me.

中文翻译:

谢谢，Lenny。感谢邀请我。

[00:02:21] Lenny Rachitsky

English:

First, let me give a little context on this conversation that we're having. I think it's going to be a little bit unique. I was doing a deep dive on the job market in tech, and I saw something that was really surprising to me that the product owner role was the third-fastest growing role in tech, and this was just in the US, the data I was looking at, but I think it's probably true broadly. This was extremely surprising to me because I've never worked with a product owner, I don't hear anyone in my circles talking about product owners, I've never wanted to hire a product owner, and it feels like it's just this very different part of the tech ecosystem that you don't hear a lot about on podcasts like this, and it's clearly growing so I felt like it'd be really helpful to spend some time helping people and helping me understand this part of the world.

(00:03:08):

I asked you to come on to talk about this. You've been up close and personal with a lot of companies working with this way of working with product owners, Scaled Agile and all these things, so I couldn't think of a better person to have on here to help us understand what's happening here, and also just to help people do this better. Melissa, thank you again for coming on and helping us understand this.

中文翻译:

首先，让我为这次对话提供一些背景。我觉得这会是一次比较独特的谈话。我之前在深入研究科技行业的就业市场，发现了一件让我非常惊讶的事情：产品负责人（Product Owner）是科技行业中增长速度第三快的职位。虽然我看的只是美国的数据，但我认为在全球范围内可能也是如此。这让我极其意外，因为我从未与产品负责人合作过，我的圈子里也没人谈论这个角色，我也从未想过要招聘产品负责人。感觉这就像是科技生态系

统中一个非常不同的部分，在像我们这样的播客中很少听到，但它显然在增长。所以我觉得花点时间帮助大家和我自己理解这个领域会很有帮助。

(00:03:08):

我邀请你来谈谈这个。你曾近距离接触过许多采用这种模式（产品负责人、大规模敏捷等）的公司，所以我认为没有比你更合适的人选来帮我们理解这里发生的事情，并帮助人们做得更好。Melissa，再次感谢你过来帮我们解惑。

[00:03:28] Melissa Perri

English:

Yeah, I'm excited to talk about this. I have been really passionate about this topic for many years and I've been talking about it in both agile circles and product management circles, so pretty excited for the listeners to hear what else is going on out there.

中文翻译:

是的，我也很兴奋能聊这个。多年来我一直对这个话题充满热情，我一直在敏捷圈和产品管理圈讨论它，所以很高兴能让听众们了解外面的世界正在发生什么。

[00:03:43] Lenny Rachitsky

English:

This episode is brought to you by Pendo, the only all-in-one product experience platform for any type of application. Tired of bouncing around multiple tools to uncover what's really happening inside your product? With all the tools you need in one simple to use platform, Pendo makes it easy to answer critical questions about how users are engaging with your product, and then turn those insights into action. Also, you can get your users to do what you actually want them to do.

(00:04:10):

First, Pendo is built around product analytics, seeing what your users are actually doing in your apps so that you can optimize their experience. Next, Pendo lets you deploy in-app guides that lead users through the actions that matter most. Then Pendo integrates user feedback so that you can capture and analyze what people actually want. The new thing in Pendo, session replays, a very cool way to visualize user sessions. I'm not surprised at all that over 10,000 companies use it today. Visit Pendo.io/lenny to create your free Pendo account today and start building better experiences across every corner of your product. Yes, you want to take your product-led know-how a step further, check out Pendo's lineup of free certification courses led by talk product experts and designed to help you grow and advance in your career. Learn more and experience the power of the Pendo platform today at Pendo.io/lenny.

中文翻译:

本期节目由 Pendo 赞助，它是唯一适用于任何类型应用程序的一站式产品体验平台。厌倦了在多个工具之间跳来跳去，只为弄清楚产品内部到底发生了什么吗？Pendo 将你所需的所有工具整合到一个简单易用的平台中，让你轻松回答关于用户如何使用产品的关键问题，并将这些洞察转化为行动。此外，你还可以引导用户完成你真正希望他们执行的操作。

(00:04:10):

首先，Pendo 围绕产品分析构建，查看用户在应用中的实际行为，以便优化他们的体验。其次，Pendo 允许你部署应用内指南，引导用户完成最重要的操作。然后，Pendo 整合了用户反馈，以便你捕捉和分析人们的真实

需求。Pendo 的新功能——会话重放（session replays），是一种非常酷的可视化用户会话的方式。今天有超过 10,000 家公司使用它，我一点也不意外。访问 Pendo.io/lenny 立即创建免费账户，开始在产品的每个角落构建更好的体验。如果你想进一步提升产品驱动（product-led）的专业知识，请查看 Pendo 由顶级产品专家主讲的免费认证课程，旨在帮助你在职业生涯中成长和进步。立即在 Pendo.io/lenny 了解更多并体验 Pendo 平台的强大功能。

[00:05:06] Lenny Rachitsky

English:

I'm excited to chat with Christina Gilbert, the founder of OneSchema, one of our long-time podcast sponsors. Hi Christina.

中文翻译:

很高兴能与 OneSchema 的创始人 Christina Gilbert 聊天，她是我们的长期赞助商之一。你好，Christina。

[00:05:13] Christina Gilbert

English:

Yes, thank you for having me on, Lenny.

中文翻译:

是的，谢谢你邀请我，Lenny。

[00:05:15] Lenny Rachitsky

English:

What is the latest with OneSchema? I know you now work with some of my favorite companies like Ramp, [inaudible 00:05:22], and Watershed. I heard that you just launched a new product to help product teams import CSVs from especially tricky systems like ERPs.

中文翻译:

OneSchema 最近有什么新动态？我知道你们现在正与我最喜欢的一些公司合作，比如 Ramp 和 Watershed。我听说你们刚刚推出了一款新产品，帮助产品团队从 ERP 等特别棘手的系统中导入 CSV 文件。

[00:05:31] Christina Gilbert

English:

Yes. We just launched OneSchema FileFeeds, which allows you to build an integration with any system in 15 minutes as long as you can export a CSV to an SFTP folder. We see our customers all the time getting stuck with hacks and workarounds, and the product teams that we work with don't have to turn down prospects because their systems are too hard to integrate with. We allow our customers to offer thousands of integrations without involving their engineering team at all.

中文翻译:

是的。我们刚刚推出了 OneSchema FileFeeds，只要你能将 CSV 导出到 SFTP 文件夹，它就能让你在 15 分钟内构建与任何系统的集成。我们经常看到客户被各种临时方案和变通方法困扰，而与我们合作的产品团队不必

因为对方系统太难集成而拒绝潜在客户。我们让客户能够提供数千种集成，而完全不需要工程团队参与。

[00:05:53] Lenny Rachitsky

English:

I can tell you that if my team had to build integrations like this, how nice would it be to be able to take this off my roadmap and instead use something like OneSchema, not just to build it, but also to maintain it forever.

中文翻译:

我可以告诉你，如果我的团队必须构建这样的集成，如果能把它从我的路线图中拿掉，转而使用 OneSchema 之类的工具，不仅是构建，还能永久维护，那该有多好。

[00:06:05] Christina Gilbert

English:

Absolutely, Lenny. We've heard so many horror stories of multi-day outages from even just a handful of ad records. We are laser-focused on integration reliability to help teams end all of those distractions that come up with integrations. We have a built-in validation layer that stops any bad data from entering your system, and OneSchema will notify your team immediately of any data that looks incorrect.

中文翻译:

绝对如此，Lenny。我们听过太多因为几条广告记录就导致持续数天停机的惨痛故事。我们专注于集成的可靠性，帮助团队消除集成过程中出现的所有干扰。我们有一个内置的验证层，可以阻止任何错误数据进入你的系统，OneSchema 会立即通知你的团队任何看起来不正确的数据。

[00:06:24] Lenny Rachitsky

English:

I know that importing incorrect data can cause all kinds of pain for your customers and quickly lose their trust. Christina, thank you for joining us, and if you want to learn more, head on over to oneschema.co, that's oneschema.co.

(00:06:39):

Before we get into the history, is there just anything broadly that you think might be helpful to share before we dive into the history of the product owner role and all these things?

中文翻译:

我知道导入错误数据会给客户带来各种痛苦，并迅速失去他们的信任。Christina，感谢你加入我们。如果你想了解更多，请访问 oneschema.co。

(00:06:39):

在我们深入探讨历史之前，关于产品负责人角色及其相关背景，你觉得还有什么大方向的内容值得分享吗？

[00:06:46] Melissa Perri

English:

Maybe it'll help to set some context of where did I see all of this start emerging as well. When I started in tech, I was very much a product manager, never heard of the product owner role before in my life. In Escaping the Build Trap I talk a lot about how we used Scrum when I started working with this team in a startup, and it was the first time I ever heard of it. That was 2011. At the time, the person who was teaching me about agile was very like, "Hey, this is flexible. We're just going to break things into sprints. We're going to sit there and actually talk about the work. This is made for us to actually get better at our jobs." We were pretty sold on it and it was never dogmatic. As I worked at other companies, I found that they were being a little more dogmatic with their Scrum, with their stand-ups, how we actually run things.

(00:07:36):

Then I started speaking at conferences, and one of the first conferences I spoke at in New York City was called Lean UX, and there was a bunch of people from the agile world there too. I learned that this was much bigger than what we were learning in my company and what we were doing in these companies. There's this whole group of people out there practicing agile, and I was like, "Oh, this is cool. I want to learn how to do things better. Teach me about your philosophies." I ended up going to a ton of agile conferences and speaking about product management.

(00:08:04):

At the time, people were really excited to hear more about it, and I started to learn that there was this product owner role in Scrum where I was talking more about how we traditionally talk about product management, understand your customer, go out test things, make sure there's a hypothesis, don't just blindly build what you want to build. I found out that that was not the case in a lot of these companies who are adopting Scrum and introducing a product owner role, so I started doing a lot of trainings through my school product institute and I'd get called into these large companies, all these large banks, probably around 2014, 2015, to help them learn product management. I was really excited about this because before, they didn't want anything to do with it. They were like, "I don't know what product management is. I don't need this."

(00:08:47):

I go in to train people and I found that a lot of them had been going through an agile transformation and they had all of these new product owners where they came in and they basically said, "Hey, you're a product owner now. Your whole role has changed." They came from all different backgrounds. Some were developers, so a lot were business people who worked on the banking side, a lot were business analysts, some were project managers, but they just collectively took a bunch of people and said, "Tada, today you're going to be a product owner because we're going to do agile now." I will come in and help train these people.

(00:09:20):

What I found was that there was a really big misconception about what those people should be doing compared to what they teach in agile and Scrum versus what we all consider great product management. I've been trying to fill that gap for the last almost 10 years working with these companies here. Then I would go to their leaders and at the beginning of these agile transformations, I'd be like, "You can't just do Scrum. That's not going to make you amazing at delivering products. There's so much more to this." The leaders didn't quite understand that.

(00:09:51):

I'm noticing this really big shift in the industry where we're finally getting there. A lot of companies are doing it well now. Capital One is a great example that took their agile transformation and started adding product management on it, and they've really turned that around in the card business. So many organizations are still at the beginning of this journey and they're at the place where I saw people 10 years ago. I think there's still a lot of companies out there. Maybe we take it for granted in tech or in Silicon Valley about how many companies are doing this and how big this scope is where they're making these roles, but they're not really doing product management end to end. That's where I've seen all of these areas and I've been trying to help organizations for the last 10 years really set up robust product management practices. It's not just one piece of development, it's how do we actually build better products?

中文翻译:

也许设定一下背景会有所帮助，看看我是从哪里开始看到这一切浮现的。当我刚进入科技行业时，我是一名纯粹的产品经理，这辈子从未听说过“产品负责人”这个角色。在《逃离构建陷阱》中，我谈到了 2011 年我在一家初创公司工作时，我们是如何使用 Scrum 的，那是我第一次听说它。当时，教我敏捷的人非常随和，他说：“嘿，这很灵活。我们只是把工作分成冲刺（sprints）。我们会坐下来讨论工作。这是为了让我们更好地完成工作。”我们非常认可这一点，而且它从不教条。但随着我在其他公司工作，我发现他们对 Scrum、站会以及运行方式变得越来越教条。

(00:07:36):

后来我开始在会议上演讲，我在纽约参加的第一个会议叫 Lean UX，那里也有很多敏捷圈的人。我意识到这比我们在公司学到的和做的要大得多。外面有一大群人在实践敏捷，我想：“噢，这很酷。我想学习如何做得更好。教教我你们的哲学。”于是我参加了大量的敏捷会议，并发表关于产品管理的演讲。

(00:08:04):

当时，人们非常兴奋能听到更多相关内容，我也开始了解到 Scrum 中有“产品负责人”这个角色。而我当时谈论的更多是我们传统意义上的产品管理：了解客户、进行测试、确保有假设、不要盲目构建。我发现很多采用 Scrum 并引入产品负责人角色的公司并非如此。于是我开始通过我的学校 Product Institute 进行大量培训。大约在 2014、2015 年左右，我被邀请到这些大公司（主要是大银行）帮助他们学习产品管理。我当时很兴奋，因为以前他们根本不想碰这个，他们会说：“我不知道什么是产品管理，我不需要这个。”

(00:08:47):

我去培训这些人，发现他们中的许多人正在经历“敏捷转型”，突然多了一堆新的产品负责人。公司基本上是说：“嘿，你现在是产品负责人了，你的整个角色都变了。”这些人背景各异：有些是开发人员，很多是银行端的业务人员，很多是业务分析师，有些是项目经理。公司只是集合了一群人说：“哒哒！今天起你们就是产品负责人了，因为我们要开始搞敏捷了。”我就去培训这些人。

(00:09:20):

我发现，关于这些人应该做什么，在敏捷/Scrum 的教学内容与我们公认的优秀产品管理之间存在巨大的误解。在过去的近 10 年里，我一直试图为这些公司填补这一空白。我会去找他们的领导者，在敏捷转型初期告诉他们：“你不能只做 Scrum。那不会让你在交付产品方面变得出色。这里面还有很多东西。”但领导者们当时并不太理解。

(00:09:51):

我注意到行业正在发生巨大的转变，我们终于快达到了。现在很多公司做得很好。第一资本（Capital One）就是一个很好的例子，他们进行了敏捷转型，并在此基础上加入了产品管理，这彻底改变了他们的信用卡业务。但仍有许多组织处于这一旅程的起点，处于我 10 年前看到的水平。我想外面还有很多这样的公司。也许在科技界或硅谷，我们理所当然地认为很多公司都在这样做，但实际上很多公司虽然设立了这些角色，却并没有真正做到端到端的产品管理。这就是我看到的现状，过去 10 年我一直在努力帮助组织建立稳健的产品管理实践。这不仅仅是开发的一个环节，而是关于我们如何真正构建更好的产品。

[00:10:42] Lenny Rachitsky

English:

I love this example of Capital One. It can work really well and you can get to a place where it's actually really productive. There's a few ways we can go about this. Do you think it would be helpful to talk through the history of the product owner role just like where it initially emerged?

中文翻译:

我很喜欢 Capital One 这个例子。它确实可以运作得很好，并达到非常高效的状态。我们可以从几个方面展开。你觉得聊聊产品负责人角色的历史，比如它最初是怎么出现的，会有帮助吗?

[00:10:56] Melissa Perri

English:

Yeah, I think that's a great place to start. I think it brings a lot of context too to what's happening. People forget about the history here. When I explain it to people, I say we had product managers in Silicon Valley, right? They were in Google, they were in all of these companies, Amazon, and they were born out of this business role. From a software native company, your software is the business, right? It's what you sell, it's what you actually look at. Our product managers in Silicon Valley, they're doing market research, they're talking to customers, they're working with developers, they're iterating, they're doing the end-to-end product management.

(00:11:31):

What happened on the other side of things, especially in large companies, is the emergence of product management from Scrum, from product ownership. That's usually the first time these companies were introduced to product management was from implementing a product owner role and then going, "Hey, we're still not meeting our goals. Are we building the right thing?" Then they started thinking product management. Where that role came from is Scrum. If we go back and talk about the history of agile, agile was a movement that got started by software developers. In 2001, the Agile Manifesto was written. A bunch of developers got together in Park City, Utah, they were all skiing together and they said, "Hey, we've been independently all working on how to develop software better."

(00:12:18):

Some people were practicing Scrum as we know it today. That was Ken Schwaber, Jeff Sutherland. There were people who were doing different types of agile frameworks as well, Kanban, where you were moving it through a Kanban board. There was behavioral-driven development, there was feature-driven development. That was the style of agile. There was XP, extreme programming, that was started by Ron Jeffries. All of these people found each other saying, "Hey, we've been trying to push the boundaries of how do we develop better software," and they got together and they wrote the Agile Manifesto as we know it today.

(00:12:51):

The Agile Manifesto is really just a guideline on how they're striving to not just be people who code what people want, but building better products. How do we build better products through software development? The premise to remember this is, and I keep saying it, but they're all software developers. Nobody was a product manager who went to that meeting. Nobody who wrote the Agile Manifesto was actually a product manager. I've spent a long time talking to these people as well over the past 10 years just saying, "Hey, how did this come about? Where did this come from?"

(00:13:23):

The one person who was really close to them who was a product manager was Jeff Patton, but he never signed the Agile Manifesto, he wasn't at that meeting. He talked to them a lot, he was able to see what was going on, but all of this was a purge from how do we build better products from a development perspective. That's really important to know. Two of the people who signed the Agile Manifesto, Jeff and Ken, as I was talking about, they were independently coming up with Scrum on their own in their different companies, and they got together and started to codify it and they said, "This is how I'm doing it, this is how I'm doing it." They ended up writing the Scrum Guide. The Scrum Guide is what a lot of people base their agile practices off of today. In the Scrum Guide, it outlines a bunch of roles that you would do on the development team, and then it says how you should be developing products.

(00:14:11):

Most people out there are working in Scrum today, and what they say is, "Let's break things down into two-week sprints." You can change the length of your sprint if you want to, but two-week sprints is pretty standard at the beginning. We define what we're going to work on in the backlog. It's the product owner's responsibility to define what goes into the backlog, write down the user stories for it, do all that. Then the development team comes in, they discuss it, the product owner prioritizes it, they ask questions, and then the development team commits to what they want to build and they go out and do it. At the end, the result is a potentially shippable product, not necessarily shippable, but potentially shippable.

(00:14:48):

They're trying to break it down into small chunks and build things instead of what they had been doing in a lot of companies, which was building stuff for three years and then releasing it in a big bang. What all of the people who signed the Agile Manifesto realized was if we do it the other way, if we do a waterfall type environment, agile waterfall, that's where we go across, there's a lot of risk because we don't test it with the customer and we don't get feedback on it if we spend three years building it and never show it to somebody. It really approached a different way of building software. It said let's chunk it down and try to get feedback faster. Really noble intention.

(00:15:24):

In the Scrum guide though, it introduces these new roles. We have developers as we know and love them, we also have a product owner, then we have a Scrum master, and the Scrum master is in charge in Scrum of actually helping people do Scrum better. That's literally their job. How do I do Scrum better? How do I make sure that the team is working well together? They host things like retrospectives where at the end of a sprint we say what went well, what didn't go well, how should we actually inspect and adapt our process. The product owner is where things get murky.

(00:15:57):

The product owner in general first showed up with Scrum, and if you go and you read the first Scrum guide, which I pulled up and started reading, because I've been very fascinated about how this is described, it says that the product owner is responsible for maximizing the value of work done, the team does the work. Interesting, because now the product owner is not quite part of the team. The team consists of developers with all the skills to turn the product owner's request into the potentially shippable increment each sprint. The team is usually seven plus or minus two members.

(00:16:29):

Then when you go further into the first version of the Scrum Guide, it does say that the Scrum master works with the customers and management to identify and instantiate a product owner. The Scrum master teaches the product owner how to do his or her job in order to optimize the value of the use of Scrum. If they don't, the Scrum master is held accountable. Then it's got another tip if we go deeper into this. It says per commercial development, the product owner may be the product manager. For in-house development efforts, the product owner could be the user department manager.

(00:17:01):

What's interesting is that that was the first version of the Scrum Guide, and I get into arguments about the Scrum Guide with people all the time. 2013's version though, the more updated one that you could go and find is the one that almost every company has run an agile transformation off of. It loses that thing that says the product manager could be the product owner. It doesn't say it anywhere in that guide. This was the first version, and you can kind of tell it was an aside. It's like, "Oh, by the way, the product owner in Scrum doesn't need to be a product manager, it could be the customer, it could be a developer." It's usually the customer.

(00:17:34):

When they were writing this too, sometimes the customer was an internal person at a bank or somewhere where we were building software who was asking for the software. They were like, "Go build me an internal tool. Go do this." Now we're just asking for requirements inside a company, and that's where you can start to see how the product owner role kind of evolved into somebody going to ask, "Hey, what do you want me to build? What's required here?," and then just listening to somebody come back and say, "I need this feature, I need this feature, I need this feature." Scrum doesn't describe how to get the stuff into the backlog, and it didn't in the 2013 manuals. The manuals have all been a little bit better, they've all kind of been updated since then, and they do describe it has to start with the vision a little bit more, you have to break down the vision for the product and get in there, but none of that existed in the early versions of Scrum.

(00:18:29):

When people got trained on how to be a product owner, what was happening here is, and this is the whole other world of Scrum over here, when people get trained to be a product owner, it's usually a two-day class where they teach them, "Hey, this is how you break down a backlog. This is how you do stand-ups with your teams. This is how you think about prioritizing work. This is how you manage your backlog, prioritize it for the developers. This is how you work with the retrospectives," but it doesn't teach them about experimentation, it doesn't teach them about market research, it doesn't teach them about data, it doesn't teach them about any of the things that we need to be a product manager.

(00:19:06):

Then what happened was we went into these agile transformations at these companies and they said, "Hey, let's adopt Scrum because Scrum was built as a way to build better products faster." It's literally the tagline. Everybody was like, "Yeah, I want to build products faster. Okay, great. Let's do Scrum." All these large organizations back in the early 2010s, in the 2000s said, "Oh, we got to be better at software. How do we do this better? Otherwise, we're going to lose when it comes to innovation." They adopted Scrum as a way to build software faster.

(00:19:42):

Now, what happened is in order to do Scrum, Scrum basically sells training. That's what Scrum does. All of these agile coaches would come in and teach the product owners, newly minted product owners, took all those people, made them into product owners, put them through a two-day class and then say, "Go." That was the beginning of all the agile transformations, and that's where a lot of companies still are today. This product owner role did not emerge from product management as we know it today, it was a way to help the developers prioritize what to work on, but that was it. The product owner was held accountable for making sure that they were working on the most pressing things or the highest value things, they do say that, but to me, if you look at it from a developer perspective, it's also the person where you can say, "Hey, well, you told me to build that, right? We didn't build it wrong. You told me to build that."

(00:20:35):

It almost gets into consulting territory where you're like, "Okay. If the product owner prioritized all this stuff for me and told me what to do, I can't be held accountable if it was the wrong thing to build." Some of that stuff does come up in a lot of teams that struggle to adopt agile, to adopt Scrum. I feel like there's a big misunderstanding out there about what is this role and what should we be doing, but the premise of this is when we talk about Scrum, it's just one piece of the puzzle, and when people talk about agile now, they almost always associate it with Scrum.

(00:21:10):

I was actually Googling agile methodologies, and like I said, the other ones, Kanbans is an agile methodology, XP is an agile methodology. They don't have product owners, they do not exist in those methodologies. There are four developers to work on things, or teams to work on things. XP would consider product managers in the teams as far as I know it, but Scrum kind of sees it as a separate thing. Agile methodologies, everybody says, "Oh, they're Scrum now," so it gets a bad connotation out there about what to do with it. I think Scrum if you do it well is not bad, but you have to understand that it's just one piece of building great products, it's not the whole thing, and companies will adopt it like it's going to radically transform everything. To be fair, a lot of times it's sold that way too.

中文翻译:

是的，我觉得这是一个很好的切入点。它为现状提供了很多背景。人们往往忘记了这段历史。当我向人们解释时，我会说硅谷一直有产品经理，对吧？他们在谷歌、亚马逊等所有这些公司里，他们诞生于业务角色。对于一家原生软件公司来说，软件就是业务，对吧？它是你销售的东西，是你真正关注的东西。硅谷的产品经理做市场调研、与客户交谈、与开发人员合作、进行迭代，做的是端到端的产品管理。

(00:11:31):

而另一边发生的事情，特别是在大公司，是产品管理从 Scrum、从“产品负责人”角色中浮现出来的。这些公司通常是通过实施产品负责人角色才第一次接触到产品管理，然后发现：“嘿，我们还是没达到目标。我们构建的东西对吗？”接着他们才开始思考产品管理。而那个角色源自 Scrum。如果我们回溯敏捷的历史，敏捷是由软件开发人员发起的一场运动。2001 年，《敏捷宣言》问世。一群开发人员在犹他州帕克城聚会，他们一起滑雪并说：“嘿，我们一直在独立研究如何更好地开发软件。”

(00:12:18):

有些人当时就在实践我们今天所知的 Scrum，比如 Ken Schwaber 和 Jeff Sutherland。还有人在做不同类型的敏捷框架，比如看板（Kanban），通过看板来推进工作。还有行为驱动开发（BDD）、特性驱动开发（FDD）。那是敏捷的不同流派。还有 XP（极限编程），由 Ron Jeffries 发起。所有这些人聚在一起说：“嘿，我们一直在尝试突破如何开发更好软件的界限。”于是他们聚在一起写下了我们今天所知的《敏捷宣言》。

(00:12:51):

《敏捷宣言》实际上只是一个指南，说明他们如何努力不只是做“按要求写代码”的人，而是构建更好的产品。如何通过软件开发构建更好的产品？要记住的前提是（我一直在强调）：他们全是软件开发人员。参加那次会议的人里没有一个是产品经理。写《敏捷宣言》的人里也没有一个是真正的产品经理。在过去的 10 年里，我也花了很多时间与这些人交谈，问他们：“嘿，这是怎么回事？这从哪儿来的？”

(00:13:23):

唯一一个与他们关系密切的产品经理是 Jeff Patton，但他从未在《敏捷宣言》上签字，也没参加那次会议。他经常与他们交流，能看到发生了什么，但这一切都是从开发的角度出发，探讨如何构建更好的产品。这一点非常重要。签署《敏捷宣言》的其中两人，Jeff 和 Ken，正如我提到的，他们在各自的公司独立构思 Scrum，然

后聚在一起将其规范化，说：“我是这么做的，你是那么做的。”他们最终编写了《Scrum 指南》。今天很多人的敏捷实践都是基于《Scrum 指南》的。指南中概述了开发团队中的一系列角色，并说明了应该如何开发产品。

(00:14:11):

今天大多数人都在使用 Scrum，他们说：“让我们把工作分解成两周一个冲刺。”你可以根据需要更改冲刺长度，但两周是标准做法。我们在待办事项（backlog）中定义要做的内容。产品负责人的职责是定义待办事项中的内容，编写用户故事（user stories）等等。然后开发团队介入讨论，产品负责人确定优先级，他们提问，接着开发团队承诺要构建的内容并开始执行。最后的结果是一个“潜在可交付的产品增量”——不一定真的要交付，但具备交付的潜力。

(00:14:48):

他们试图将其分解成小块来构建，而不是像很多公司以前那样，闭门造车三年，然后搞个“大爆炸”式发布。所有签署《敏捷宣言》的人都意识到，如果采用另一种方式，即瀑布流（waterfall）环境，风险会非常大，因为如果你花三年时间构建却从未展示给任何人看，你就无法在客户身上进行测试，也得不到反馈。它实际上提出了一种不同的软件构建方式：拆分任务，尝试更快获得反馈。初衷非常高尚。

(00:15:24):

但在《Scrum 指南》中，它引入了这些新角色。我们有熟知的开发人员，还有产品负责人，然后是 Scrum Master。Scrum Master 的职责是帮助人们更好地执行 Scrum。这真的是他们的工作：如何把 Scrum 做得更好？如何确保团队协作良好？他们主持回顾会（retrospectives），在冲刺结束时讨论哪些做得好，哪些不好，以及如何检查和调整流程。而“产品负责人”这个角色就开始变得模糊了。

(00:15:57):

产品负责人最初出现在 Scrum 中。如果你去读第一版《Scrum 指南》(我为了研究特意翻了出来)，它说产品负责人负责“最大化团队所做工作的价值”。有趣的是，当时产品负责人并不完全被视为团队的一部分。团队由开发人员组成，他们拥有将产品负责人的请求转化为每个冲刺中潜在可交付增量的所有技能。团队通常是 7 加减 2 人。

(00:16:29):

再深入看第一版指南，它确实提到 Scrum Master 与客户和管理层合作，以识别并实例化一个产品负责人。Scrum Master 教导产品负责人如何履行职责，以优化 Scrum 的使用价值。如果没做好，Scrum Master 要负责。再往深看，它还有一个提示：对于商业开发，产品负责人可以是产品经理；对于内部开发，产品负责人可以是用户部门的主管。

(00:17:01):

有趣的是，那是第一版《Scrum 指南》，我经常为此与人争论。然而，2013 年的版本（几乎所有公司进行敏捷转型时参考的更新版本）删掉了“产品经理可以是产品负责人”这句话。指南里再也找不到了。在第一版中，这看起来像是个顺带一提的备注：“噢，顺便说一下，Scrum 里的产品负责人不需要是产品经理，可以是客户，也可以是开发人员。”通常就是客户。

(00:17:34):

在编写这些内容时，有时客户是银行或某处的内部人员，他们要求开发软件，比如：“给我做个内部工具。”现在我们只是在公司内部询问需求，这就是产品负责人角色演变过程的开端：有人去问“嘿，你想让我建什么？这里有什么要求？”，然后听对方说“我需要这个功能，我需要那个功能”。Scrum 并没有描述如何将内容放入待办事项，2013 年的手册里也没有。之后的手册有所改进，描述了必须从愿景开始，分解愿景，但在早期版本的 Scrum 中，这些都不存在。

(00:18:29):

当人们接受产品负责人培训时，情况是这样的（这是 Scrum 的另一个世界）：通常是一个为期两天的课程，教他们“嘿，这是如何分解待办事项，这是如何与团队开站会，这是如何思考工作优先级，这是如何管理待办事项为开发人员排期，这是如何参与回顾会”。但它不教实验，不教市场调研，不教数据，不教任何我们认为作为产品经理所需的东西。

(00:19:06):

接着，这些公司开始了敏捷转型，他们说：“嘿，让我们采用 Scrum 吧，因为 Scrum 的口号就是‘以更快速度构建更好产品的方法’。”每个人都想快点。于是，在 2010 年代初和 2000 年代，所有这些大型组织都说：“噢，我们必须擅长软件开发，否则我们会在创新上输掉。”他们把 Scrum 当作加速软件开发的手段。

(00:19:42):

然而，为了实施 Scrum，Scrum 实际上是在“卖培训”。这就是 Scrum 的商业模式。所有这些敏捷教练会进来培训那些新上任的产品负责人，把一群人变成 PO，让他们上两天课，然后说：“去吧。”这就是所有敏捷转型的开端，也是很多公司现状的缩影。这个产品负责人角色并非源自我们今天所知的产品管理，它只是帮助开发人员确定工作优先级的一种方式，仅此而已。虽然说 PO 要对“确保工作在最紧迫或最有价值的事情上”负责，但对我来说，从开发人员的角度看，PO 也是那个你可以推卸责任的人：“嘿，是你让我建那个的，对吧？我们没建错，是你让我建的。”

(00:20:35):

这几乎进入了咨询领域，你会觉得：“好吧，如果产品负责人为我排好了所有优先级并告诉我做什么，那么如果建错了东西，我也不能负责。”在许多挣扎于采用敏捷和 Scrum 的团队中，确实会出现这种情况。我觉得外界对这个角色以及我们应该做什么存在巨大的误解。但前提是，当我们谈论 Scrum 时，它只是拼图的一块，而现在人们谈论敏捷时，几乎总是把它等同于 Scrum。

(00:21:10):

我之前在谷歌搜索敏捷方法论，正如我所说，看板（Kanban）是敏捷，XP 也是敏捷。但那些方法论里没有产品负责人，这个角色在那些方法论里根本不存在。它们只有开发人员或团队来处理工作。据我所知，XP 会把产品经理视为团队的一部分，但 Scrum 把 PO 看作一个独立的存在。现在大家都说“敏捷就是 Scrum”，这让敏捷在外面背负了一些不好的名声。我认为 Scrum 如果做得好并不坏，但你必须明白它只是构建伟大产品的一个环节，而不是全部。公司往往把它当作能彻底改变一切的神药，说实话，很多时候它也是被这么推销的。

[00:21:59] Lenny Rachitsky

English:

There's a bigger picture question that's coming to mind as you talk about this. I'm imagining founders listening to this and smaller companies listening to this be like, "Why do we need any of this?", especially Silicon Valley startups, "We're just going to build stuff. We don't need these frameworks, we don't need a Scrum master. We just have awesome developers and product managers and we're just going to build awesome stuff." I don't know anyone that has worked this way that has built amazing things. Can you talk a bit about who ends up looking for solutions here, where this even comes from, what companies need help here versus, "I just don't need any of this?"

中文翻译:

听你聊这些，我脑子里浮现出一个更宏观的问题。我能想象初创公司的创始人们听了会说：“我们为什么需要这些？”，尤其是硅谷的初创公司，“我们直接开干就行了。我们不需要这些框架，不需要 Scrum Master。我们有优秀的开发和 PM，我们直接做牛逼的东西就行。”但我没见过哪个以这种（教条）方式工作的团队做出了惊人的产品。你能谈谈到底是谁在寻找这些解决方案？这到底是从哪儿来的？什么样的公司需要这些帮助，而什么样的公司可以大方地说“我根本不需要这些”？

[00:22:36] Melissa Perri

English:

A lot of large companies turn to Scrum or to the frameworks, and it's because they traditionally didn't grow up building software. They're looking at how do I implement something that has rigor at scale, and that's where you see a lot of Scrum come up. Now, I've seen startups using Scrum. Some of them do it fine, they understand that it's just more about trying to get things out the door every two weeks to test it with customers. I think if you keep that philosophy, like I said, I used it and we didn't have a lot of rigor around it, that was fine. When we were doing Scrum, when I did it with my team back in OpenSky, we got to this point where we were like, "Two weeks is too long. We're just going to ship things every week."

(00:23:12):

We just talked to each other, we skipped stand-ups, which is sacrilege in Scrum, but we kipped daily stand-ups. We didn't need to stand around and talk about it, we talked to each other every day. For me, what was amazing and where I see teams actually thrive when they start using Scrum is when you go and talk to people. You're having the conversations about the work, you're breaking it down, you're understanding it, so the developers and the rest of the team can go hit the road running and people can ask important questions. If you're not doing that, that's where I think things like a framework help, but if you already are doing that, you don't need a framework, you don't need Scrum, you don't need to be prescribed to this two-week sprint or anything like that.

(00:23:52):

As long as you have a methodology, it doesn't even have to be a defined methodology. As long as you have a way of working that gets things out to customers, well, who cares? Who actually cares? Where there's a lot of, I think, baggage in the industry and where I hear product managers get really frustrated and other people as well, developers too, is that when you do Scrum by the book or how people teach it and how they write about it, it's a million meetings. I know they were put in there so that people were forced to talk, but when you already know what you're supposed to work on, why do you need to keep doing meetings? Shouldn't you just go do some work?

(00:24:28):

A lot of developers complain, a lot of product managers complain that Scrum has too many meetings and they don't actually get to do work. That's where I think you have to go back to the inspect and adapt part. A lot of people who are very religious about Scrum will come and yell at me about this. They're like, "Oh, well, that's not how it's supposed to be. You're supposed to inspect and adapt." Agree, but a lot of people aren't doing it and that's the piece where you go and you say, "Is this serving us? If not, let's get rid of it. Let's not do those types of things." When you're a small startup, I don't think you need a lot of this overhead. It's really much designed for larger scale companies, and those are the ones that you see really adopting it.

中文翻译:

很多大公司转向 Scrum 或这些框架，是因为他们传统上不是靠开发软件起家的。他们在寻找如何在规模化的情况下实施具有严谨性的东西，这就是为什么你会看到很多 Scrum 的出现。我也见过初创公司使用 Scrum，有些做得不错，他们明白这只是为了每两周把东西推出去给客户测试。我觉得如果你秉持这种理念，就像我说的，我以前用的时候并没有那么多条条框框，效果也很好。当我和我在 OpenSky 的团队做 Scrum 时，我们后来觉得："两周太长了，我们每周发一次货吧。"

(00:23:12):

我们直接互相交流，跳过了站会（这在 Scrum 里简直是大逆不道），但我们确实跳过了每日站会。我们不需要站在一起讨论，因为我们每天都在交流。对我来说，Scrum 真正让团队蓬勃发展的地方在于你真正去和人交

谈。你们在讨论工作、分解任务、理解需求，这样开发人员和团队其他成员就能迅速上手，人们也能提出重要问题。如果你没做到这一点，那么框架之类的东西会有所帮助；但如果你已经在这么做了，你就不需要框架，不需要 Scrum，也不需要被规定死两周一个冲刺。

(00:23:52):

只要你有一套方法论——甚至不需要是定义明确的方法论——只要你有一套能把东西交付给客户的工作方式，谁在乎呢？谁真的在乎？我认为行业中存在很多包袱，让产品经理、开发人员和其他人都感到沮丧的地方在于：当你照本宣科地执行 Scrum，或者按照别人教的那样去做时，会有开不完的会。我知道设立这些会议是为了强迫人们交流，但如果你已经知道该做什么了，为什么还要一直开会？难道不该去干活吗？

(00:24:28):

很多开发人员和 PM 抱怨 Scrum 会议太多，导致没时间真正干活。这就是我认为必须回到“检查与调整”（inspect and adapt）这一步的原因。很多 Scrum 的虔诚信徒会为此对我大喊大叫：“噢，那不是它本来的样子，你应该检查并调整。”我同意，但很多人并没这么做。你应该问问：“这对我们有帮助吗？如果没有，那就把它砍掉。不要做那些形式主义的事情。”如果你是一家小型初创公司，我认为你不需要这么多管理开销。这主要是为大规模公司设计的，也是那些公司真正采用它的原因。

[00:25:06] Lenny Rachitsky

English:

From what I've seen on the data, it's also companies that are as you I think alluded to at the beginning, not necessarily software first, product first companies. Feels like it's very common in banks and telecom companies and companies that aren't product First and software first.

中文翻译:

从我看到的数据来看，正如你开头提到的，这些公司通常不一定是“软件优先”或“产品优先”的公司。感觉在银行、电信公司以及那些非产品/软件驱动的公司中非常普遍。

[00:25:21] Melissa Perri

English:

There are SaaS companies that do Scrum out there and they like it, and I don't think they're very dogmatic about it.

中文翻译:

也有一些 SaaS 公司在做 Scrum 并且很喜欢它，而且我觉得他们并不教条。

[00:25:27] Lenny Rachitsky

English:

Got it. Yeah.

中文翻译:

明白了。是的。

[00:25:28] Melissa Perri

English:

They do it for a reason of just trying to provide some more context to their teams about how to work together at scale. I've also seen places where they don't prescribe whatever methodology you want to work with for the teams, but instead they'll spend a lot more effort breaking down the road maps, thinking about what are we going to do each quarter, trying to set those themes, and then they just let the teams run, and that works as well. I think it really depends how they adopt it, but I would say it's not a hard and fast rule that no startups are doing this. Some are doing it, I just don't know how it's going for them. To me, it might be overkill if you're doing that with a team that's pretty experienced in doing this.

中文翻译:

他们这样做是为了给团队提供更多关于如何在大规模协作的背景。我也见过一些地方，他们不规定团队必须使用哪种方法论，而是花更多精力分解路线图（roadmaps），思考每个季度要做什么，设定主题，然后让团队自由发挥，这也很有效。我认为这取决于他们如何采用。我不会说初创公司绝对不做这个，有些在做，只是我不知道效果如何。对我来说，如果你的团队经验非常丰富，这样做可能有点大材小用（overkill）。

[00:26:12] Lenny Rachitsky

English:

I was insinuating is less just Scrum as a general idea and more very structured rigid processes and also product owners. Then there's this whole concept of SAFe that we can talk about. Should we get into that or should we talk about product manager versus product owner and just the challenges people have there?

中文翻译:

我刚才暗示的不仅仅是作为通用概念的 Scrum，更多是指那些非常结构化、僵化的流程，以及“产品负责人”这个角色。还有 SAFe 这个完整概念我们可以聊聊。我们是先聊这个，还是先聊产品经理与产品负责人的区别以及人们面临的挑战?

[00:26:33] Melissa Perri

English:

Let's talk about SAFe because that's where a lot of this started to get Confusing

中文翻译:

我们聊聊 SAFe 吧，因为那是很多事情开始变得混乱的地方。

[00:26:33] Lenny Rachitsky

English:

Okay, cool. Let's go there. Yeah.

中文翻译:

好，太棒了。我们聊聊这个。

[00:26:37] Melissa Perri

English:

Scaled Agile Framework came out of the desire to figure out how do we scale Scrum and different processes and bring it to organizations at scale. It originated from a more structured approach to agile too called Rational Unified Processing. Now, SAFe wasn't the first thing that started at scale. There was also LeSS, which is a Scaled Agile Framework, and then Jeff Sutherland who did Scrum has Scrum@Scale. It's not the only scaling framework out there. There's a lot, there's actually a lot out there, but SAFe was one that was marketed the best. The way it's marketed is will tell you everything you need to do, to do all of your agile teams with Scrum and put them all together.

(00:27:22):

The idea behind SAFe was that Dean Leffingwell came up with it. He wanted to really show how you tie multiple teams together at scale in an organization and how do you bring some rigor and process to that. The executives at really large enterprises, we're talking tens of thousands of people, they love SAFe because it prescribes a lot of an operating model of what to do when it comes to development, but it also gets billed as like, "Hey, this is the whole model for you to go do software." If you look at it, it's a big map that everybody kind of makes fun of a little bit, and it describes all different things on it if you look at the map. You can click in and you can see the definitions and you can see what's going on in the areas.

(00:28:10):

The SAFe image has gotten bigger and bigger over time. I think, what is this? Version six. I do know a lot of people who worked on SAFe, so I know a lot of trainers and I've worked with companies. The first time I was introduced to SAFe was when I was working with a bank back in 2015. I came in to train their product managers, I'm doing my training, we're setting them up on how to go talk to customers, talk about hypotheses, MVP, and somebody came up to me and they said, "Hey, Melissa, all of this is great, but I don't have time to go talk to customers because I'm a product owner." I was like, "Well, what are you doing on a day-to-day basis? What don't you have time for it?", "I got to write my user stories." I'm like, "Okay. How many user stories do you write per day?"

(00:28:53):

This was for the developers to have a full backlog so they could all work, right? She's like, "Oh, yeah. I spend pretty much 40 hours a day writing user stories." I'm like, "On what?" We're like, "What are you controlling?" She's like, "The login API for a bank." I'm like, "Can you log in?" She's like, "Yeah," and I'm like, "So what else are you working on? Is there a new initiative? Is there a new thing?" It's like, "No, I was reorganized into a team where I became the product owner. I have a product manager who goes and talks to customers, but then she comes and she tells me what to build, and then I write the user stories around it and I put it into the backlog for the teams." I was like, "What is this?" Then they said, "That's SAFe. This is what we're working towards."

(00:29:34):

This was my first experience with SAFe, and then I ran into another company that did it, another company, same thing over and over and over again, where all these product owners were just basically trying to keep these backlogs full for developers, and they were working on such a narrow level. When a lot of organizations too I saw it reorganized into agile teams, they did it by component. Everybody was over every tiny little feature, and these teams were massive, super huge scope, and some of the stuff was just not prioritized. It was done, you didn't need to work on it. They were finding work to do so people wouldn't get fired. That's how the product owners operated. There was all this legacy baggage sometimes in companies where they were all re-put on things by component, and they're just making up work to do.

(00:30:21):

SAFe introduced this kind of split between product manager and product owner, and if you look at the map, the product owner is part of the agile team where they sit with the Scrum masters, which is a team coach that they call here and the developers, and then the product manager is sitting with a system architect and what they call a release train engineer. What SAFe does is they pull a bunch of agile teams into a release train and you get on the train when you're ready to ship things and you make sure that it all goes pretty smoothly to get to that potential shippable increment or that big feature launch that you would be doing with SAFe.

(00:30:58):

SAFe's really good at prescribing how to do that, they're great at describing how to do the release trains, how to bring those teams together, how to put them on it. Then they do this thing called big room planning where they get the entire release train together, all these teams, they put them in a room and in every quarter you're breaking down what we're all going to work on. Where I hear frustration from teams every time I come in and train them is that when you do big room planning, a lot of times it's a commitment. You start at the quarter, they haven't been doing good discovery because remember, these people have not been trained on good discovery so they don't really know what they should be working on, they haven't been out talking to customers a lot of times. They kind of scramble, they figure out what needs to happen. Usually they have a backlog of stuff that does need to happen, it just has to get done. They map it all out in a big room together, they commit to it, and then that's the quarter.

(00:31:47):

They ask me, "When am I supposed to do discovery?" I'm like, "Well before that ideally you should have a vision. You should be breaking it down, you should be putting discovery into that vision, talking to customers, feeding that in there." Then I hear, "We don't have time to do that because we sprint back to back." I was like, "What does that mean?", and they're like, "As a team, we go and we basically do two-week sprint into two-week sprint into two-week sprint, and I got to make sure my developers are full. I got to make sure they have things to work on. If I go take time off to go talk to customers, which also is not my job as a product owner, it's my product manager's job, they'll feed me in what the customers are saying, then I break those down into features and I can work with the developers on it." That's how all of this stuff starts going.

(00:32:32):

What happens in organizations that they don't understand here is that it's not the most efficient way to work. I see a lot of developers out there become almost ... How do you describe it? Reliant on the product manager or product owner to tell them what to do. Even though you build them this great vision and you explain what needs to happen, they go, "Oh, I can't work on it because the product owner hasn't prioritized it." Then they asked me, "If I don't have enough for the developers to do on feature work, what are they supposed to do?" I said, "I guarantee you there's a ton of tech debt they could be working on. You don't have to scope that out. Let them choose what's the most important thing. They should be working together as developers and architects to figure out how to tackle some of that tech debt, how to get into it while you're figuring out is this the right thing to be building."

(00:33:23):

With all of this stuff, people feel like they don't have time because they're in a million meetings, and the expectations of these companies is that every sprint, we're delivering software towards these roadmaps that we promised in the last quarter and we're not checking to see if they're right, we're not checking to see if they're actually helping us move it forward, and a lot of times the organizations are not set up with the right feedback mechanisms, the right user research and the right data to tell us if everything is working so they can feed that in to the next release planning. They're just planning, planning, planning, breaking it down into sprints and going.

(00:33:58):

SAFe is not good at describing how you do all that other work. In a lot of this stuff too, there's pieces that they put onto this map of SAFe where they're like, "Hey, you should do OKRs," and it's like, "This is what OKRs are. You should do a roadmap. This is what a roadmap is." How all of that cycle works together where you're balancing discovery and delivery and feeding it in is really confusing in organizations. Then what it's basically saying is a lot of the discovery work goes into the product managers, and the product managers, the product owners report into the product managers. What I've seen that doesn't work here is that you're basically making these product owners order-takers. They are extremely tactical, and then when it's time to actually be more strategic, let's say you want to be promoted to a product manager, some organizations, that's not even the same business line, not even the same career path. It's product owners go over here and product managers go here and they report into different people.

(00:34:58):

If you ever want to move from product owner to product manager, a lot of times you don't get experience with the strategy, figuring out what customers want, breaking it down, looking at the market research, determining is this valuable, is this what we should be working on. They're not even getting exposure or a chance to do that because SAFe is like, "No, that's the product manager's job. Your job is to go really deep and work with the developers."

中文翻译:

大规模敏捷框架（SAFe）源于人们想要弄清楚如何扩展 Scrum 和不同的流程，并将其带入大型组织。它也起源于一种更结构化的敏捷方法，称为“理性统一过程”（Rational Unified Processing）。SAFe 并不是第一个做规模化的，还有 LeSS（大规模敏捷框架），Scrum 的创始人 Jeff Sutherland 也有 Scrum@Scale。它不是唯一的，实际上有很多，但 SAFe 是营销做得最好的。它的营销方式是：它会告诉你做所有 Scrum 敏捷团队并把他们整合在一起所需的一切。

(00:27:22):

SAFe 的理念是由 Dean Leffingwell 提出的。他想展示如何在一个组织中大规模地将多个团队联系在一起，以及如何为此引入严谨性和流程。超大型企业（我们说的是成千上万人的规模）的高管们喜欢 SAFe，因为它规定了一套关于开发的运营模式，而且它被宣传为：“嘿，这就是你做软件的完整模型。”如果你看它的图，那是一张巨大的地图，大家经常拿它开玩笑，地图上描述了各种不同的东西。你可以点击进去看定义，看各个区域发生了什么。

(00:28:10):

SAFe 的图表随着时间的推移变得越来越大。我想现在是第六版了。我认识很多研究 SAFe 的人，认识很多培训师，也和很多公司合作过。我第一次接触 SAFe 是在 2015 年与一家银行合作。我去培训他们的产品经理，教他们如何与客户交谈、讨论假设、做 MVP。有人走过来对我说：“嘿，Melissa，这一切都很好，但我没时间去和客户交谈，因为我是产品负责人（PO）。”我问：“那你日常都在做什么？为什么没时间？”她说：“我得写用户故事。”我问：“那你每天写多少个用户故事？”

(00:28:53):

这是为了让开发人员有一个完整的待办事项列表，这样他们就能一直有活干，对吧？她说：“噢，是的。我每天几乎花 40 个小时（夸张说法）写用户故事。”我问：“写什么？”她说：“银行的登录 API。”我问：“现在能登录吗？”她说：“能。”我问：“那你还在写什么？是有新计划吗？”她说：“不，我被重新分配到一个团队当 PO。我有一个产品经理（PM）去和客户谈，然后她回来告诉我建什么，我再围绕它写用户故事，放进团队的待办事项里。”我当时就惊了：“这是什么操作？”他们说：“这就是 SAFe，这是我们的目标。”

(00:29:34):

这是我第一次接触 SAFe，后来我遇到一家又一家公司，情况一模一样：所有的 PO 基本上只是在努力填满开发人员的待办事项，他们的工作范围极其狭窄。当很多组织重组为敏捷团队时，他们是按“组件”划分的。每个人负责一个小功能，团队庞大，范围却极小，有些东西根本不需要优先级，已经做完了，但他们为了不让大家失业，得找活干。这就是 PO 的运作方式。公司里有很多历史包袱，按组件划分，大家只是在编造工作。

(00:30:21):

SAFe 引入了 PM 和 PO 之间的这种分裂。如果你看地图，PO 是敏捷团队的一部分，与 Scrum Master（他们称之为团队教练）和开发人员坐在一起；而 PM 则与系统架构师和所谓的“发布火车工程师”（Release Train Engineer）坐在一起。SAFe 的做法是把一堆敏捷团队拉进一列“发布火车”，当你准备好发布时就上车，确保一切顺利达到潜在可交付增量。

(00:30:58):

SAFe 非常擅长规定如何执行这些：如何运行发布火车、如何整合团队。然后他们搞一个叫“大房间规划”（Big Room Planning）的东西，把整列火车的团队都聚在一个房间里，每个季度分解大家要做的工作。我每次去培训时听到的抱怨是：当你做大房间规划时，很多时候那是一种“承诺”。季度开始时，他们还没做过好的“探索”（discovery），因为这些人没受过相关培训，不知道该做什么，也没怎么和客户谈过。他们手忙脚乱地弄清楚需要发生什么，通常只是把必须完成的待办事项列出来。大家在一个大房间里把计划画出来，承诺执行，这个季度就定死了。

(00:31:47):

他们问我：“我什么时候做探索？”我说：“理想情况下，在那之前你应该有愿景，分解愿景，把探索放进愿景里，谈客户，把反馈喂进去。”然后我听到：“我们没时间，因为我们的冲刺是一个接一个的。”我问什么意思，他们说：“作为一个团队，我们就是两周接两周地冲刺，我得确保开发人员的工作是满的。如果我请假去谈客户（而且 PO 觉得那不是自己的活，是 PM 的活），PM 会把客户说的喂给我，我再分解成功能给开发。”这就是一切变味的开始。

(00:32:32):

组织不明白的是，这并不是最高效的工作方式。我看到很多开发人员变得……怎么说呢，变得依赖 PM 或 PO 告诉他们做什么。即使你给他们描绘了宏伟愿景，解释了需要发生什么，他们也会说：“噢，我不能做，因为 PO 还没排优先级。”然后 PO 问我：“如果我没有足够的功能开发给开发人员做，他们该做什么？”我说：“我保证有成堆的技术债可以处理。你不需要去定义那个，让他们自己选最重要的。开发和架构师应该合作解决技术债，而你则去弄清楚我们构建的东西是否正确。”

(00:33:23):

因为这些，人们觉得没时间，因为他们陷在无数的会议中。公司的预期是每个冲刺都要交付软件，朝着上个季度承诺的路线图前进，但我们不去检查这些是否正确，不去检查是否真的在推进业务。很多时候，组织没有建立正确的反馈机制、用户研究和数据支持，无法告诉我们一切是否奏效，从而无法为下一次发布规划提供参考。他们只是在不停地计划、计划、计划，分解成冲刺，然后执行。

(00:33:58):

SAFe 不擅长描述如何做那些“其他工作”。在 SAFe 的地图上，他们丢了一些碎片，比如“你应该做 OKR”、“你应该做路线图”。但这些环节如何协同工作——如何平衡探索与交付并形成闭环——在组织中非常混乱。它基本上是说，大部分探索工作由 PM 做，而 PO 向 PM 汇报。我看到的失败之处在于，你基本上把 PO 变成了“接单员”。他们极其战术化。当到了需要变得更有战略眼光的时候，比如你想晋升为 PM，在某些组织里，这甚至不是同一条业务线，也不是同一个职业路径。PO 在这边，PM 在那边，汇报给不同的人。

(00:34:58):

如果你想从 PO 转到 PM，很多时候你根本没有制定战略、弄清客户需求、分解市场调研、判断价值的经验。你甚至没有机会接触这些，因为 SAFe 说：“不，那是 PM 的活。你的活是深入细节，和开发人员待在一起。”

[00:35:21] Lenny Rachitsky

English:

Wow, okay. A lot of this sounds quite absurd as someone hearing all of the details and looking at this image. That being said, many companies are adopting this. It feels like it's growing. More and more companies are adopting this as the way to work. I imagine the incentive is we just want to build great software and we don't know exactly how, and there's this process we can plug in and it'll help us do it. I guess thoughts on that, and do you find it can work or often works or often doesn't work? What is your experience with people adopting this and how it goes?

中文翻译:

哇，好吧。作为一个听完细节并看着这张图的人，这听起来相当荒谬。话虽如此，许多公司正在采用它。感觉它还在增长，越来越多的公司把它当作工作方式。我猜动机是：我们只想构建出色的软件，但不知道具体怎么做，而这里有一个可以“即插即用”的流程，能帮我们实现。你对此怎么看？你觉得它有效吗？是经常奏效还是经常失败？你看到的采用情况和结果是怎样的？

[00:35:59] Melissa Perri

English:

I know a ton of companies that adopted SAFe about eight years ago and have gotten rid of it. Capital One just came out and said they got rid of all their agile roles, all their Scrum roles. They were early adopter of SAFe, they don't do it anymore. They wrote about that in the newspaper. I've seen it happen more often. Now, in a lot of our organizations too I'll see parts of them do SAFe and other parts not do SAFe. It could change business line to business line. I don't think though that people grasp how much it's still out there. I get questions on SAFe every single day on the podcast. Everybody asks me, "Why are we still doing this?" It's for what you said, executives buy SAFe because it's the only framework out there that basically draws them a map and says, "Plug and play, do this."

(00:36:50):

That's why everybody's so excited about it because it's the only thing that specifies things to this level, and they went, "Oh, it's something I can understand, it's something that actually has definitions around it." To be fair, that was a great thing for SAFe to do as a marketing tool. Bravo, they created this thing that everybody wants, a good product to sell, but it's overkill, and that's what I keep hearing from organizations is it's basically taking the responsibility away from leaders to go figure their stuff out themselves as well. If you are a new leader and you've just been dropped into this role, I have tremendous empathy for them because yeah, where do you get started? How do you try to run a technology organization?

(00:37:33):

Somebody came and told me, the CIO came and told me I'm in charge now. I'm in charge of all of the developers, or I'm in charge of all the product managers. Now, where do you start? I can totally tell why people adopt SAFe because you're like, "Oh, I've been looking for the handbook. I've been looking for something to do here." The problem is it's only solving a little bit of the puzzle, which is bringing those teams together. People do say it does really strains well, but it doesn't tell you also how to do your job as a leader, it leaves it all out. They talk about portfolio visions and portfolio management and SAFe there too, but more often than not, I come in and I find everybody above product owners and product managers, let's talk about directors of product, VPs of product, they don't know what they should be doing as a VP of product or a director of product. It's like, "What's my role? What should I be feeding in here?"

(00:38:23):

SAFe doesn't even have that in there, that's not even a role. Product manager going up into those levels is not really there. What do you do when you own a whole product line in an organization? What you do when you're the head of product for a credit card at a bank, right? What's my job? Doesn't say that. There's a lot of people out there in these organizations that I've been working with who I'm like, "You are supposed to be doing strategy, and this is how you do strategy. This is how you go out and talk to customers. This is the patterns that we have in software. Are you doing a platform strategy? Do you need APIs? How do you think about your app strategy, rolling it out? How do you do this here?" All of that stuff doesn't quite come from ways of working, which is what SAFe is doing. It's about how do you do your jobs in those areas?

(00:39:12):

A lot of organizations who adopt SAFe don't realize that you need a head of product, you need somebody to actually be feeding that vision all the way down and make sure it's breaking up around the teams and controlling that portfolio vision and doing all of these things into it. I have not seen SAFe slowing down by any means out there for people adopting it, I see more and more organizations adopting it. I think we take for granted too in Silicon Valley how many people are just starting on their journey for digital transformation. There's a lot of pharmaceutical companies, banks, insurance companies, they outsource their development or they had an IT team, but they never had to really think about it before because digital wasn't as important. Now they do.

(00:39:59):

Some of these companies, most of these companies are Fortune 50 companies, right? Fortune 100 companies. I think a lot of the ones I see, at least banks, realized early on, "Hey, when it comes to apps and how people interact with our stuff, software is important," but there's a lot of companies that did not catch that train and they're just starting, and then they turn to things like SAFe because it gives them a guideline. "Hey, I've never done this before. I've been in this bank for 40 years. All I know is waterfall type development. What do we do?" Then we'll go, "We'll go look at SAFe."

中文翻译:

我认识一大堆大约八年前采用 SAFe 但现在已经把它废了的公司。Capital One 最近刚宣布裁撤了所有的敏捷角色和 Scrum 角色。他们曾是 SAFe 的早期采用者，现在不做了，报纸上都登了。我看到这种情况越来越频繁。现在，在很多组织里，我会看到一部分部门在做 SAFe，另一部分不做，这可能因业务线而异。但我认为人们没意识到它在外面还有多大的市场。我在播客里每天都会收到关于 SAFe 的问题，每个人都问我："为什么我们还在做这个？"原因正如你所说：高管们买 SAFe，是因为它是市面上唯一一个能给他们画出一张地图并说"即插即用，照着做就行"的框架。

(00:36:50):

这就是为什么大家这么兴奋，因为它是唯一一个能细化到这种程度的东西。他们会觉得："噢，这是我能理解的东西，它有明确的定义。"公平地说，作为营销工具，SAFe 做得很棒。太赞了，他们创造了一个人人都想要的好产品来卖。但它太重了（overkill），我从各组织听到的反馈是，它基本上剥夺了领导者自己去解决问题的责任。如果你是一个刚被空降到这个职位的新领导，我非常同情他们，因为确实，你从哪儿开始呢？你如何尝试运行一个技术组织？

(00:37:33):

有人跑来告诉我，CIO 跑来告诉我，现在我负责了，负责所有的开发人员或所有的产品经理。现在，你从哪儿开始？我完全能理解为什么人们采用 SAFe，因为你会觉得："噢，我一直在找手册，我一直在找该做点什么。"问题是，它只解决了拼图的一小部分，即把团队整合在一起。人们确实说它在"发布火车"方面做得很好，但它没告诉你作为领导者该如何工作，它把这部分全漏了。虽然 SAFe 也谈论组合愿景（portfolio visions）和组

合管理，但通常情况下，我进去发现 PO 和 PM 之上的人——比如产品总监、产品 VP——他们不知道作为 VP 或总监该做什么。他们会问：“我的角色是什么？我该往这里面喂什么？”

(00:38:23):

SAFe 里甚至没有这些，那甚至不是一个角色。PM 往上晋升到这些级别的路径在里面并不清晰。当你负责组织中的整条产品线时，你该做什么？当你是一家银行信用卡的负责人时，你的工作是什么？它没说。在这些组织中，我遇到很多人，我会告诉他们：“你应该做战略，这是做战略的方法。这是你如何出去和客户谈。这是我们在软件中拥有的模式。你在做平台战略吗？你需要 API 吗？你如何考虑应用策略并推广？你在这里怎么做？”所有这些东西并不完全来自“工作方式”（ways of working），而 SAFe 做的只是工作方式。它没教你如何在这些领域履行职责。

(00:39:12):

很多采用 SAFe 的组织没意识到你需要一个产品负责人（Head of Product），需要有人真正将愿景自上而下传递，确保它在团队中分解，控制组合愿景并处理所有这些事情。我没看到 SAFe 的采用率有任何放缓的迹象，反而看到越来越多的组织在采用它。我认为在硅谷，我们也把很多事情想得太理所当然了，其实有很多公司才刚刚开始数字化转型。有很多制药公司、银行、保险公司，他们以前外包开发或者只有个 IT 团队，以前从不需要考虑这些，因为数字化没那么重要。现在重要了。

(00:39:59):

这些公司大多是财富 50 强或 100 强。我想我看到的很多公司，至少是银行，很早就意识到：“嘿，涉及到 App 和用户交互时，软件很重要。”但还有很多公司没赶上那趟车，现在才开始，于是他们转向 SAFe 之类的东西，因为它提供了指南。“嘿，我以前没做过这个，我在银行待了 40 年，我只知道瀑布流开发。我们该怎么办？”然后他们就会说：“我们去看看 SAFe 吧。”

[00:40:33] Lenny Rachitsky

English:

I love that we spend time on that because I think it's really important. You can be cynical about all this and be like, "What the hell are people thinking? This is crazy," but as you described, people just have a problem to solve, they've never done this before, they look for solutions, they find something that seems right, they see other people doing this and like, "Okay, let's try this thing." What you've seen is it rarely actually works out the SAFe specific approach.

(00:40:59):

There's a few ways I think we can help folks. One is someone trying to do say an agile transformation or a digital transformation, your advice for how to actually do that better. Then I want to talk about say you're a product owner or a PM within an organization that works like this. What can you do? Maybe let's start with the first. Say someone's trying to figure out, "We need to build better products. Something's not working right." SAFe is an option. Your suggestion is don't maybe do that. What should people do? I know you're not going to have the answer in a short answer, but generally, how should people approach this?

中文翻译:

我很喜欢我们花时间聊这个，因为这真的很重要。你可以对这一切冷嘲热讽，觉得“这些人到底在想什么？这太疯狂了”，但正如你所描述的，人们只是有待解决的问题，他们以前没做过，于是寻找解决方案，找到了一个看起来对的东西，看到别人也在做，就说：“好吧，试试这个。”而你看到的是，SAFe 这种特定的方法很少能真正奏效。

(00:40:59):

我觉得我们可以从几个方面帮助大家。一是对于那些尝试进行敏捷转型或数字化转型的人，你对如何做得更好的建议。然后我想聊聊，如果你是这种组织内部的 PO 或 PM，你能做什么？也许我们先从第一个开始。假设有人在想：“我们需要构建更好的产品，有些地方不对劲。”SAFe 是一个选项，而你的建议是“也许别做那个”。那人们该怎么做？我知道你不可能用一句话回答，但总的来说，人们应该如何切入？

[00:41:33] Melissa Perri

English:

Yeah. When I've worked with companies on digital transformations, you want a development operating model. That's where a lot of these agile methodologies came out of. You have to understand that's just the development operating model, that's not actually going to help you with go-to-market, with launching your products and with product management. What I advise for companies to do is first sit down and say, "Hey, how do we think about building our operating model?" When I think of product operating models and what I do with companies is we break out how do you determine product strategy? Do you have a good product strategy? You look at your organizational design. How are we actually organized around our products? Do we have good coverage of product managers and do we have skilled product managers up and down the organization?

(00:42:20):

Then we want to look at product operations. Do we have the infrastructure in there to help support these teams? Can they get the data to make decisions? Can they actually be in touch with customers? A lot of these large organizations haven't actually thought through many of those steps as well that enable product managers and development teams to be successful. They don't have ways for them to go and talk to customers. That's why they're not doing it. I have a lot of empathy for people in these organizations as well who can't do product management well because of the bureaucracy or the things around it. Leaders need to solve that, right? They need to understand what the role is and they need to open it up.

(00:42:57):

Then we got to look at our culture and incentives. Are we just rewarding people for shipping as many things as possible, which is like, "Hey, just put everything you possibly can into that release train or that backlog," or are we coming back and saying, "Hey, is this valuable? Is this tying it back to our business?" Many organizations do not have a great product strategy, many large organizations that I've worked with, and it's that tying it back to the value piece, tying it back to, is this going to reach our company goals? If you are a huge organization and let's say making billions of dollars a year, and your goal is expand geographically, what are you doing in your portfolio to actually enable that? What products are you building to expand geographically? So many organizations don't have the transparency to actually even see that.

(00:43:50):

One crazy thing, a lot of people give large organizations a lot of flack for, and I know Marty does this too, for focusing on processes. I don't think processes are the enemy here. For example, if I hear somebody really worried about getting a roadmapping tool in there or something like that, I'm like, "Yes, you need that because you have no idea what your 4,000 teams are doing." If they're actually coming back to the business goals, you have no infrastructure in there to be able to see that transparency. Those types of blocking tackling is absolutely necessary for a transformation for a organization to be stood up around software product management. You have to have the transparency to actually see those things.

(00:44:29):

You do need to have enough process so that you as an organization can be efficient in getting things out the door, and that's what I think SAFe was trying to do, but it's not working because it's not solving the problems of the product management and it's not solving that problem of connecting the value back to the product teams. Instead, it's seen as a role that almost babysits developers or tells everybody what to do. Where's the discovery? Where do those things come in? I know with the SAFe image that we got over here, they try to drop things like Lean UX in there, which Jeff Gothelf thinks it's hilarious, but it's not really pulling it all together of how do we do this on a cadence? How do we help people go out there and actually talk to customers? How do we enable them to do it?

(00:45:15):

If you're starting a transformation, it's not just thinking about how do we build the product, but you should also be thinking about how do we launch the product and how do we make sure this is the right product to do. That's the big pieces of it, and that's where all that product strategy comes in. You should also look at the career paths. This is what really bothers me about agile transformations and what bothers me with Scrum and SAFe is that when we organized in these large organizations into agile teams, we made all these new roles called a product owner, and so many organizations don't have a career path for them so they email me and they go, "What's my career path? What do I do next? Where am I supposed to go?"

(00:45:59):

I've been saying for 10 years, this is not a team role, it's not just a team role, it's a business role and it rolls all the way up to helping you further your business. You have to make sure that people on teams can be promoted to running multiple teams, can be promoted to running an entire product line. To us, that's so simple in Silicon Valley native software companies, but it's still unheard of in other organizations. What happens too, and this is where I think leadership and C-suite needs to really pay attention, because we're transforming in this way of working, what happens is some of the roles that we had before do not serve us now. Maybe we don't need a million project managers, maybe people in the business who decided what we're going to build, are they the right people to bring with us on this next phase into product management? Can they learn? Can they grok software? Do they understand those pieces? That's what we have to ask.

(00:46:58):

Organizations are so afraid sometimes to put these career ladders in because it kind of overhauls their traditional ways of working, and then they've got people who've been in these organizations for 40 years and now you're saying, "Hey, you're actually not in charge of that, the product manager is in charge of that," and that's scary. A lot of them get in the way because of that. If you really want to transform though, the C-suite has to be like, "Hey, we're going in this direction," and just put it down because I've seen it run by a lot of middle managers, a transformation run by tons of middle managers, and those are the jobs that are usually in most jeopardy when you start transforming and you have to re-skill and you have to figure out what to do, and they don't want to do that. They're not going to be the ones who jump up and down and say, "Hey, let's do this."

(00:47:41):

There's a lot of people out there, I think, pushing organizations to try harder and to internally as well. I've worked with a lot of people who run these transformations who just really want it to work, and I think they do it with the best of intentions, but the C-suite has to understand this is not just a transformation project, this is a whole new way of working, and if we want a whole new way of working, we have to really rise to that occasion.

中文翻译:

是的。当我与公司合作进行数字化转型时，你需要一个“开发运营模型”。这就是很多敏捷方法论的来源。但你必须明白，那只是开发运营模型，它并不能真正帮助你进行市场推广、产品发布和产品管理。我建议公司做的第一件事是坐下来问：“嘿，我们如何考虑构建我们的运营模型？”当我思考产品运营模型时，我会和公司一起拆解：你如何确定产品战略？你有好的产品战略吗？看看你的组织设计：我们是如何围绕产品进行组织的？我们是否有足够的产品经理覆盖？组织上下是否有经验丰富的产品经理？

(00:42:20):

然后我们要看产品运营（Product Operations）。我们是否有基础设施来支持这些团队？他们能获得做决策所需的数据吗？他们能真正接触到客户吗？很多大组织并没有仔细考虑这些能让 PM 和开发团队成功的步骤。他们没有让团队去和客户交谈的途径，所以团队就不去做。我也非常同情这些组织里的人，因为官僚主义或周围的环境，他们无法做好产品管理。领导者需要解决这个问题，对吧？他们需要理解这个角色是什么，并为其扫清障碍。

(00:42:57):

接着我们要看文化和激励机制。我们是仅仅奖励那些交付东西最多的人吗（比如“嘿，把所有能塞的东西都塞进发布火车或待办事项里”），还是我们会回头问：“嘿，这有价值吗？这是否与我们的业务挂钩？”我合作过的许多大组织都没有出色的产品战略，关键在于将其与价值挂钩，与公司目标挂钩。如果你是一个年收入数十亿美元的庞大组织，目标是地域扩张，那么你的产品组合中做了什么来支持这一点？你正在构建什么产品来扩张？很多组织甚至没有透明度来看到这些。

(00:43:50):

有一件疯狂的事，很多人（我知道 Marty Cagan 也是）批评大组织过于关注流程。我不认为流程是敌人。例如，如果我听到有人非常担心引入路线图工具之类的东西，我会说：“是的，你需要那个，因为你根本不知道你那 4000 个团队都在干什么。”如果他们真的要回归业务目标，你必须有基础设施才能看到那种透明度。这种“基本功”（blocking and tackling）对于一个围绕软件产品管理建立起来的组织转型是绝对必要的。你必须拥有能看到这些东西的透明度。

(00:44:29):

你确实需要足够的流程，以便组织能高效地交付产品，这正是 SAFe 试图做的，但它没奏效，因为它没解决产品管理的问题，也没解决将价值连回产品团队的问题。相反，它被看作是一个几乎是在“看护”开发人员或告诉每个人该做什么的角色。探索（discovery）在哪儿？这些东西从哪儿切入？我知道在 SAFe 的图里，他们尝试塞进 Lean UX 之类的东西（Jeff Gothelf 觉得这很滑稽），但它并没有真正把“我们如何按节奏进行”、“我们如何帮助人们走出去谈客户”、“我们如何赋能他们”这些事情整合在一起。

(00:45:15):

如果你开始转型，不仅要考虑如何构建产品，还要考虑如何发布产品，以及如何确保这是正确的产品。这是核心部分，也是产品战略发挥作用的地方。你还应该关注职业路径。这正是我对敏捷转型、Scrum 和 SAFe 感到困扰的地方：当我们在大组织中组建敏捷团队时，我们创造了“产品负责人”这个新角色，但很多组织没有为他们提供职业路径。所以他们给我发邮件问：“我的职业路径是什么？我下一步该做什么？我该往哪儿走？”

(00:45:59):

我说了 10 年了，这不仅仅是一个团队角色，它是一个业务角色，它一直延伸到帮助你推进业务。你必须确保团队中的人可以晋升到管理多个团队，晋升到管理整条产品线。对硅谷的原生软件公司来说，这很简单，但在其他组织中仍然闻所未闻。此外（这也是领导层和高管需要真正关注的地方），因为我们正在改变工作方式，以前的一些角色现在已经不再适用了。也许我们不需要一百万个项目经理，也许以前决定我们要建什么的业务人员，并不是带我们进入产品管理下一阶段的合适人选。他们能学习吗？他们能理解软件吗？他们理解这些环节吗？这是我们必须问的问题。

(00:46:58):

组织有时非常害怕建立这些职业阶梯，因为这会彻底颠覆传统的运作方式。你面对的是在组织里待了 40 年的人，现在你告诉他们："嘿，其实你不负责那个了，产品经理负责。"这很吓人。很多人因此成为阻碍。但如果你真的想转型，高管层必须明确："嘿，我们要往这个方向走。"并把它落实。因为我见过很多由中层管理人员主导的转型，而当你开始转型、需要重新培训技能并弄清楚该做什么时，这些人的职位通常是最危险的，他们不想这么做。他们不会是那个跳起来说"嘿，让我们干吧"的人。

(00:47:41):

我认为外面有很多人在推动组织更加努力，内部也是如此。我与很多负责这些转型的人合作过，他们真心希望成功，初衷是好的。但高管层必须明白，这不仅仅是一个转型项目，这是一套全新的工作方式。如果我们想要一套全新的工作方式，我们就必须真正迎接这个挑战。

[00:49:13] Lenny Rachitsky

English:

There's a few things I want to pull out from what you just shared. One is, just to clarify, you recommend not using SAFe, you don't think that's a good approach?

中文翻译:

我想从你刚才分享的内容中提取几点。一是，为了明确起见，你建议不要使用 SAFe，你认为这不是一个好的方法?

[00:49:21] Melissa Perri

English:

I do not recommend using SAFe. Yeah.

中文翻译:

我不建议使用 SAFe。是的。

[00:49:24] Lenny Rachitsky

English:

Great.

中文翻译:

太好了。

[00:49:25] Melissa Perri

English:

There are people who like SAFe. Let me just say this, there are people who found success with SAFe. Every single person I have talked to who like SAFe found success with SAFe, they ended up ripping it up and making it into something else. It's not actually SAFe by the book. If you do that, fine, that's any process. If you ended up adopting SAFe and you want to go back and look at it and say, "Actually let's just get rid of all the stuff that's not working and keep the stuff that is," fine, but being open to understanding this is not the way that we do good product management. There's not a lot in SAFe about doing good product management. That's the stuff that we have to understand. It could help in certain areas, and I do think it does help in certain areas, bring some rigor to things, but if you take it too far, it will destroy things.

(00:50:17):

There's actually a great story about a water company in the Netherlands and they decided to adopt SAFe, and this was on the news a couple months ago. They decided to adopt SAFe in their IT teams and start working with it. They ended up going bankrupt, and the reason they ended up going bankrupt is because the teams were learning the processes for SAFe, they were taking so long to deploy their new invoicing system and payment collections that they couldn't collect payments from customers because they got so caught up in the process.

(00:50:53):

That's what I see happen a lot in these organizations. Instead of talking about what's really important, which is, "Hey, how are we serving our customers? How are we winning in this market? How do we stem churn? How do we do all these things?", we're talking instead about, "What stand-ups are we doing? Oh, how do we do this release planning? Oh, my God, you guys didn't sprint back to back, you did it wrong." We're talking about work about work, but we're not actually getting into what are we achieving here, and that's the part I do not like about rigid processes when it comes to this.

中文翻译:

确实有人喜欢 SAFe。我得说明一下，有人通过 SAFe 获得了成功。但我交谈过的每一个喜欢 SAFe 并获得成功的人，最终都把它拆了，改成了别的东西。那实际上已经不是教科书上的 SAFe 了。如果你这么做，没问题，任何流程都可以这样。如果你采用了 SAFe，然后回头看说："其实我们把没用的都扔了，只保留有用的，"那也行。但要保持开放的心态去理解：这并不是我们做优秀产品管理的方式。SAFe 中关于如何做好产品管理的内容并不多。这是我们必须明白的。它在某些领域可能有帮助，我也确实认为它在某些方面能带来严谨性，但如果你做得太过火，它会摧毁一切。

(00:50:17):

实际上有一个关于荷兰一家水务公司的著名故事，几个月前刚上新闻。他们决定在 IT 团队中采用 SAFe。结果他们最终破产了，原因是团队忙于学习 SAFe 的流程，导致部署新的发票系统和收款系统花了太长时间，以至于他们无法向客户收款，因为他们完全陷入了流程之中。

(00:50:53):

这就是我在这些组织中经常看到的。我们不去讨论真正重要的事情，比如："嘿，我们如何服务客户？我们如何在这个市场获胜？我们如何阻止流失？我们如何做这些事？"，反而是在讨论："我们要开什么样的站会？噢，我们怎么做这个发布规划？天哪，你们没有背靠背地进行冲刺，你们做错了。"我们在讨论"关于工作的工作"（work about work），却没真正进入"我们在这里取得了什么成就"。这就是我不喜欢僵化流程的原因。

[00:51:28] Lenny Rachitsky

English:

That touches on the other theme I wanted to bring up is it feels like the stuff is a kind of replacement for skilled, talented people, a product leader that understands how to do these things and has product taste and has organized teams to build great product. It feels like people are just, "We don't have that so we're going to create this. This process is going to fix all our problems." Can you talk about just the importance of that, the people you hire to run these things as key to this, if that's true?

中文翻译:

这触及了我想提出的另一个主题：感觉这些东西像是对“有技能、有才华的人”的一种替代。比如一个懂行、有产品品味、能组织团队构建伟大产品的产品领导者。感觉人们在想：“我们没有那样的人，所以我们要创造这个流程。流程会解决我们所有的问题。”你能谈谈人的重要性吗？招聘合适的人来运行这些东西是否才是关键？

[00:52:00] Melissa Perri

English:

In a lot of organizations, the people who buy SAFe, they have not run large scale technology organizations before, or they're new to this way of working so they adopt SAFe and they hope it works because it looks like a nice plan, like we said, to go out and do things. When you're doing a transformation, a lot of companies are pulling people into these roles for the first time. I've said since day one that I've been working with companies, it's okay and I think it's noble to want to train people and put them in different roles. Cool. If you're going to spend money upskilling your people, do that, but you also have to intersperse people who know what they're doing. I think at leadership it's really important to bring in somebody who knows what they're doing to help run this type of thing.

(00:52:44):

There are more people out there and more leaders who have done this before because we've been doing this for 10 years. There's Shruti Patel, she's chief product officer at US Bank for small business banking. I just had her on the podcast. She worked at Shopify, she saw how great teams worked, and then she was able to come and help apply that at a bank. She's experienced, right? She's an experienced product person who comes in to help. Melissa Douros is the CPO of Green Dot Bank and she had worked at Discover Financial leading the transformation there, did all that work, and then could bring it to Green Dot Bank. She can see what needs to happen, what needs to actually go on here.

(00:53:23):

We've got more and more people out there who have done this before who are looking for these opportunities to do it in the bank, and I think it's important for C-suite to bring them in to actually look at that. Where I've seen transformations be the most successful in all these organizations is when you do that mix, you keep some of your people, but you also bring people in to learn. I get hired all the time to come in and train. I've worked with almost every Fortune 50 company at this point, fortune 100 company too, and I get in, I come in to train a lot of product managers. We do it through Product Institute and we'll train everybody. What used to happen about eight years ago is they train everybody and they would say, "Go." Where do you go after you bring in the consultants to do training to keep learning? How do they watch other people in the organization do great product management if there's nobody in the organization who's done it before?

(00:54:20):

Luckily I think a lot of organizations are realizing that, so more leaders are out there who are saying, "Hey, I've got to actually intersperse skills here. I need to bring in some more directors who are experienced here, some more individual contributors who are experienced here." Those organizations I think are wildly successful because they recognize it and they say, "I've got to make sure that people can learn from others." That's how you keep developing, that's how we all keep developing, it's not just doing all external classes. That's where I think these things become powerful. You could do that at all levels. You don't have to just do it with the teams, you could do it at director level, you could do it at VP level. That's how we should be thinking about this.

(00:49:57):

Now, there are some CPOs out there and some VPs of product in these large organizations who are new to this way of working, but they've committed themselves to learning and to trying to figure out how to do it best. They're not saying, "Hey, I'm just going to adopt SAFe or I'm just going to do whatever is over here," they're actually saying, "What don't I know?" I'm watching them go out to talk to other CPOs, do all these other things. They usually have great market knowledge, great business knowledge, and they're fantastic at strategy, and then they hire people underneath them who are great at the other pieces like the execution and getting the software at the door. I think those people are successful in it as well because they notice their skill gaps and they hire for it just like any great leader would.

(00:55:41):

In these organizations, I do see sometimes SAFe for something being a crutch for people who don't know what they're doing to bring in. If you really think about, "Hey, how do I make this better?", and have that continuous learning mindset and that way to want to propel this forward, I think you'll consider other options and start to think about broader than just SAFe, broader than just agile, what do we need to make this successful?

(00:56:07):

The key part of this too is recognizing that product management is not just this role in Scrum. I say this in my talk too, I say Take Scrum away. You still need product management, right? Product owner doesn't exist without Scrum, that's not a thing, but you still need product managers and that's why all product owners should be product managers, they should be fundamentally product managers. That's why I do not like these career trajectories that keep them separate. Sure, if you want to have a principal IC product manager like they do in a lot of large Silicon Valley companies, perfect, let people keep working on those things. They don't have to go into management, but that doesn't mean they're different. Between an IC product owner and an IC product manager, it shouldn't be different there.

中文翻译:

在很多组织中，购买 SAFe 的人以前没有管理过大规模技术组织的经验，或者他们对这种工作方式很陌生。所以他们采用 SAFe 并希望它能奏效，因为正如我们所说，它看起来像是一个很好的计划。在进行转型时，很多公司是第一次把人推向这些角色。从我开始与公司合作的第一天起我就说过，想要培训员工并让他们担任不同角色是好的，也是高尚的。没问题。如果你打算花钱提升员工技能，那就去做，但你还必须穿插一些“知道自己在做什么”的人。我认为在领导层，引入一个懂行的人来帮助运行这类事情非常重要。

(00:52:44):

现在外面有更多做过这件事的人和领导者，因为我们已经做了 10 年了。比如 Shruti Patel，她是美国银行（US Bank）小企业银行业务的首席产品官。我刚邀请她上了播客。她曾在 Shopify 工作，见过优秀的团队是如何运作的，然后她能把这些经验应用到银行。她很有经验，对吧？她是一个能提供帮助的资深产品人。Melissa Douros 是 Green Dot Bank 的 CPO，她曾在 Discover Financial 领导过转型，积累了经验，然后带到了 Green Dot Bank。她能预见需要发生什么，这里真正需要进行什么。

(00:53:23):

现在有越来越多做过这件事的人在寻找在银行等机构实践的机会，我认为高管层引入他们非常重要。我看到的转型最成功的组织，通常是做了这种“混合”：保留一部分老员工，但也引入新人来带动学习。我经常被聘请去进行培训，到目前为止我几乎与所有财富 50 强和 100 强公司合作过。我进去培训很多产品经理，通过 Product Institute 培训每个人。大约八年前的情况是，他们培训完所有人后就说：“去吧。”但请顾问做完培训后，员工去哪儿继续学习呢？如果组织里没有一个做过这件事的人，他们如何观察别人是如何做优秀产品管理的呢？

(00:54:20):

幸运的是，我认为很多组织正在意识到这一点。越来越多的领导者说：“嘿，我得在这里穿插一些技能。我需要引入一些有经验的总监，一些有经验的个人贡献者（IC）。”我认为这些组织非常成功，因为他们意识到：“我必须确保人们可以向他人学习。”这就是你持续发展的方式，也是我们所有人持续发展的方式，而不仅仅是上外部课程。这就是我认为这些事情变得强大的地方。你可以在各个层面这样做，不只是在团队层面，也可以在总监层、VP 层。这就是我们应该思考的方式。

(00:54:57):

现在，这些大组织中也有一些 CPO 和产品 VP 对这种工作方式比较陌生，但他们致力于学习并努力弄清楚如何做得最好。他们不是说“嘿，我就采用 SAFe 或者随便什么”，而是说“我还有什么不知道的？”我观察他们去和其他 CPO 交流，做各种尝试。他们通常有很好的市场知识和业务知识，擅长战略，然后他们聘请擅长执行和软件交付的人。我认为这些人也能成功，因为他们察觉到了自己的技能差距，并像任何优秀的领导者一样针对性地招聘。

(00:55:41):

在这些组织中，我确实有时看到 SAFe 成了那些不懂行的人的“拐杖”。如果你真的思考“我如何让它变得更好”，并拥有持续学习的心态，你就会考虑其他选项，开始思考除了 SAFe 和敏捷之外，我们还需要什么来取得成功。

(00:56:07):

这里的关键部分还在于认识到：产品管理不仅仅是 Scrum 中的这个角色。我在演讲中也说过：把 Scrum 拿掉，你仍然需要产品管理，对吧？没有 Scrum ，“产品负责人”就不存在，那不是一个独立的东西。但你仍然需要产品经理，这就是为什么所有的产品负责人（PO）都应该是产品经理（PM），他们从根本上应该是 PM。这就是为什么我不喜欢把他们分开的职业轨迹。当然，如果你想拥有像硅谷大公司那样的资深 IC 产品经理，完美，让人们继续钻研业务，他们不一定要进入管理层，但这并不意味着他们（与 PM）是不同的。在 IC 级别的 PO 和 IC 级别的 PM 之间，不应该有区别。

[00:56:55] Lenny Rachitsky

English:

Perfect segue to where I wanted to go next, which is say you are a product owner today listening to this and you're like, "Man, this is exactly my life. What can I do?", what's your advice to folks in that role right now about how to potentially become product managers, build the skills they need to not just be stuck in this career path that doesn't go anywhere?

中文翻译:

完美的过渡，这正是我接下来想聊的。假设你是一名正在听这个节目的产品负责人（PO），你心想：“天哪，这说的就是我的生活。我该怎么办？”对于目前处于这个角色的人，你有什么建议？他们如何才能转变为产品经理，建立所需的技能，而不是被困在这个没有前途的职业路径上？

[00:57:14] Melissa Perri

English:

Yeah. I think the first thing is bringing awareness to that your role is more than just working with the developers. A lot of leaders argue with me that we need product owners because it just doesn't scale. You've seen massive companies at scale where they don't have any product owners. I do not understand that. It's a weak argument to me, it's a very weak argument. It just means you don't know how to distribute the work evenly and give a little bit of strategic guidance to product owners so that they can gon, or product managers, on a team so that they can go and build visions and cut down features and stuff like that. If you're a product owner and you're like, "Hey, I don't have the opportunity to talk to customers, my product manager does that. I am just working with the teams. I want to be more strategic, I want to think longer term," I'd say try to take some ownership over that and push back on the things that are being given to you.

(00:58:04):

I was doing a workshop for Mind the Product back in the day, and I had a product owner in my workshop and she said, "I don't think that things we're working on," they were doing SAFe, "are the right things to work on." I said, "You should bring this to your manager," and she was like, "I don't know. I'm going to get fired. I don't think it's the right thing. What am I going to work on if we're not going to work on this?" I'm like, "Well, this is the beauty of product instead of project, we stay with the product." Just because your project ends doesn't mean that you lose a job. She put together this whole thing, went and said, "I don't think we should be working on this," and they promoted her. They were like, "Fantastic." She took this leap of faith and went out there and started saying, "This is more. We need to do more."

(00:58:44):

I think if you're a product owner and there's no career path for you, start asking leaders what your career path is because it's going to make them go, "Oh, great question. What should the career path be?" There's a lot of literature out there about how we make career paths, so you can start there. Ask what's next for you after this product owner role. I would ask the product managers if they're doing all the customer research, see if you can do some customer research with them. Go sit in on this. A lot of them will say, "I don't have time. I don't have time to do this." Strip back all the user stories you're working on, stop thinking about it as a quantitative metric that needs to just go up and up. Instead, really think about the value you're delivering with your team. Is this the right thing?

(00:59:24):

When you talk to leaders and when you present your case, you say it in a way of, "Hey, I'm working on X, Y, and Z feature. What's the goal here? When we release this, what do we hope will happen?" I think that's one of the best questions anybody can ask if they're worried their company is not focused on outcomes. What do we hope will happen when we release this? What metrics are we going to change? How do we instrument it to make sure that's true? Then we can go back and actually see if it changed. One simple question to get alignment on it, and then you can start to say, "Oh, that didn't work or this did work." Great. Why did it work? You can open up those conversations.

(01:00:00):

I'd say there's a lot of things you can do to help move your companies forward, and I have seen in a lot of these organizations too, a groundswell of product owners and product managers saying, "Hey, what's next for me? What's going on?" That makes the organizations go and figure it out. I was working with one Fortune 10 company not too long ago, their C-suite, I've been working with them for a very long time and they're finally like, "Hey, we're going to codify the product manager role and we're going to have it all the way up and down our organization, we're going to make roles, we're going to make responsibilities." To me, that was music to my ears, but the reason they were doing it too is because they noticed there was a lot of churn in the organization in that role, and they also realized it's a critical role.

(01:00:44):

They're losing good people because people from the outside are coming because they want to work for this great big organization that's doing super well, fantastic, but they get in there and they go, "Where's my career path? What am I supposed to do? Where am I supposed to go?" A lot of leaders are out there now realizing, "Hey, we do have to get our stuff together," and the only reason they're coming to this conclusion as well is because they're looking around seeing other people doing it and they hear it from the teams, they hear it from the product managers. I don't want people to think, "Hey, I have no power. I'm in a 10,000 person organization." The more you bring this up, the more your leaders will respect it because they don't want to lose you, they don't want to lose good people. If you want to be great at your job and you need more support there, speak up, speak up.

(01:01:31):

At a certain point, I do tell people this. If you feel like you can't do great product management in your organization, try to find another organization. I know that is hard to say and I respect people are tied to insurances and it's hard to change jobs, but if you do have the opportunity to look for another organization that does it well, I would go there. I would also say in large corporations too, I've seen certain business lines and certain divisions do it super well and then others not. If you are in a large corporation, maybe think about moving internally, laterally to a different team and seeing if you can work there. I'd find the leaders who know what they're doing and go work for that. That's usually the best move here.

中文翻译:

是的。我认为第一件事是意识到你的角色不仅仅是与开发人员合作。很多领导者跟我争论说我们需要 PO，因为（PM 模式）无法规模化。但你见过很多规模巨大的公司根本没有 PO。我不理解那种说法，对我来说那是一个非常软弱的论点。那只意味着你不知道如何均匀分配工作，不知道如何给团队中的 PM 一点战略指导，让他们去构建愿景、裁剪功能等等。如果你是一名 PO，你觉得“嘿，我没机会和客户谈，那是 PM 的事。我只是在和团队干活。我想更有战略眼光，想考虑长远”，我会说，尝试对这件事承担起所有权（ownership），并对塞给你的那些琐事进行反击。

(00:58:04):

我以前在 Mind the Product 做工作坊，有个 PO 说：“我不认为我们正在做的东西（他们当时在做 SAFe）是正确的东西。”我说：“你应该去跟你的经理谈。”她说：“我不知道，我可能会被开除。我不觉得这是对的，但如果不做这个，我该做什么？”我说：“这就是‘产品’而非‘项目’的魅力所在，我们是与产品共存的。”仅仅因为你的项目结束了，并不意味着你会失业。她整理了一整套方案，跑去说：“我不认为我们应该做这个，”结果他们提拔了她。他们觉得：“太棒了。”她迈出了那勇敢的一步，开始说：“这还不够，我们需要做得更多。”

(01:00:00):

我认为如果你是一名 PO 且没有职业路径，就开始问领导你的职业路径是什么。这会促使他们思考：“噢，好问题。职业路径应该是怎样的？”外面有很多关于如何制定职业路径的文献，你可以从那里开始。询问在这个 PO 角色之后，你的下一步是什么。我会问 PM，如果他们在做所有的客户研究，看看你是否能参与其中。去旁听。很多人会说“我没时间”。那就削减你正在写的那些用户故事，不要把它看作一个必须不断增长的量化指标。相反，真正思考你和团队交付的价值。这是正确的东西吗?

(00:59:24):

当你与领导交谈并陈述你的理由时，你可以这样说：“嘿，我正在做 X、Y、Z 功能。这里的目标是什么？当我们发布它时，我们希望发生什么？”我认为这是任何人如果担心公司不关注结果（outcomes）时能问出的最好的问题之一。发布后我们希望发生什么？我们要改变什么指标？我们如何衡量以确保它是真的？然后我们可以回头看它是否真的改变了。一个简单的问题就能达成一致，然后你可以开始说：“噢，那个没奏效”或者“这个奏效了”。太好了，为什么奏效？你可以开启这些对话。

(01:00:00):

我想说，你可以做很多事情来推动公司前进。我也在很多组织中看到，PO 和 PM 们形成了一股自下而上的力量，在问：“嘿，我的下一步是什么？发生了什么？”这迫使组织去解决问题。不久前我与一家财富 10 强公司合作，我与他们的高管层合作了很久，他们终于说：“嘿，我们要规范化产品经理角色，我们要让它贯穿整个

组织，我们要明确角色和职责。”对我来说，这简直是天籁之音。他们之所以这么做，是因为他们注意到这个角色的流失率很高，而且他们意识到这是一个关键角色。

(01:00:44):

他们正在失去优秀的人才，因为外面的人进来是因为想为这家表现优异的大型组织工作，太棒了，但进来后会问：“我的职业路径在哪儿？我该做什么？我该往哪儿走？”现在很多领导者意识到：“嘿，我们确实得把这些理顺了。”他们得出这个结论的唯一原因，是因为他们看到别人在做，而且听到了团队和 PM 的心声。我不希望人们觉得“嘿，我没有权力，我在一个万人的大组织里”。你提得越多，你的领导就越重视，因为他们不想失去你，不想失去优秀的人才。如果你想在工作中表现出色并需要更多支持，那就大声说出来。

(01:01:31):

在某种程度上，我也会告诉人们：如果你觉得在目前的组织中无法做好产品管理，那就尝试换个环境。我知道这很难说出口，我也尊重人们受限于保险等因素，换工作不容易。但如果你有机会寻找一个做得好的组织，那就去吧。我还会说，在大公司里，我也见过某些业务线和部门做得非常好，而其他的则不然。如果你在大公司，也许可以考虑内部转岗，横向移动到另一个团队看看。找到那些懂行的领导者，去为他们工作。这通常是最好的选择。

[01:02:11] Lenny Rachitsky

English:

The point you made about how a lot of companies don't have any product owners and have scaled very wide, just to reinforce that in the data dive that we did on job market trends, no tech company has a product owner basically, no top tech company. I know there's an important distinction here. These are tech, software first, product first businesses where their business is the software they're building, and a lot of the companies we're talking about here are not that. They're banks and telecoms, pharmaceutical companies. I get that it's a very different world, but I think it's important to highlight, "You can become very big." Google has no product owners as far as I know, Amazon, Microsoft, Netflix, no company you've heard of that's a tech company has a product owner. They're all product managers, they're all product managers.

中文翻译:

你提到的关于很多公司没有 PO 却能规模化得很大的观点，我想补充一点：在我们对就业市场趋势的数据研究中，基本上没有一家顶级科技公司设有“产品负责人”这个职位。我知道这里有一个重要的区别：这些是科技驱动、软件优先、产品优先的企业，软件本身就是他们的业务。而我们这里讨论的很多公司并非如此，它们是银行、电信、制药公司。我明白这是一个非常不同的世界，但我觉得强调这一点很重要：“你可以变得非常庞大（而不需要 PO）。”据我所知，谷歌没有 PO，亚马逊、微软、奈飞，任何你听过的科技公司都没有 PO。他们全是产品经理，全是产品经理。

[01:02:56] Melissa Perri

English:

Yeah. I don't want people to think that there aren't people who build great software in these large corporations too because there are. There's pockets of people who are doing it super, super well. If you are one of those people who's been pushing the boundaries, doing great work, and your title is a product owner, what I always tell people on your resume, if you're looking for your next job so that you're not pushed out, let's say, of these large corporations like a Google or somewhere like that, and that's where you want to work in a tech firm, make sure you describe how you did your job from a value perspective.

Do not talk about your agile cadences. Get Scrum out of there. Talk about what value you brought to the users and what metrics you moved, and that's how your resume should be laid out.

(01:03:40):

I do read tons of resumes to hire people, and also chief product officer, same thing. If I see immediately implemented Scrum processes across the organization, I'm like, "No, that's not what I need. That's not what I was looking for." I was looking for what are you going to do to push the strategy in that part of the organization? What are you going to do to actually build better products for customers? Then when you get into the interview, you can talk about what things you did to do that, but you want to make sure that you're focused on really understanding the customer and translating that into great products, and the outcomes that we were looking for when you do it on your resume, I think that's important.

中文翻译:

是的。我不希望人们认为大公司里就没有能做出好软件的人，因为确实有。有些小团队做得非常非常出色。如果你是那些不断突破界限、做出卓越工作的人之一，而你的头衔是 PO，我总是告诉大家：在你的简历上，如果你想找下一份工作，不想被谷歌这类大科技公司拒之门外，一定要从“价值”的角度描述你的工作。不要谈论你的敏捷节奏。把 Scrum 那些词儿拿掉。谈谈你为用户带来了什么价值，你推动了什么指标。这就是你简历应该呈现的方式。

(01:03:40):

我读过无数份简历来招人，招 CPO 也是一样。如果我一看到“在整个组织中实施了 Scrum 流程”，我会想：“不，那不是我需要的，那不是我寻找的东西。”我寻找的是：你打算如何推动该部门的战略？你打算如何为客户构建更好的产品？然后当你进入面试时，你可以谈论你为此做了什么。但你要确保你专注于真正理解客户并将其转化为伟大的产品，以及你在简历中体现出的我们所追求的结果（outcomes）。我认为这很重要。

[01:04:15] Lenny Rachitsky

English:

Okay. I want to spend more time here because this is so important. This is highlighting here's the difference between a product owner and a product manager. If you want to move into product management and become a great PM, if you're a product owner today, even if you're not a product owner and just want to get into product management, can you again just highlight here's the big difference and here's the skills that people value most in a PM versus a product owner.

中文翻译:

好，我想在这里多花点时间，因为这太重要了。这突显了 PO 和 PM 之间的区别。如果你想转入产品管理并成为一名优秀的 PM，如果你现在是 PO，或者即使你不是 PO 只是想入行，你能再次强调一下最大的区别是什么吗？以及人们在 PM 身上最看重的技能与 PO 有什么不同？

[01:04:38] Melissa Perri

English:

Yeah. When I see product owners write out their resumes or describe their job functions, they always approach it from a process standpoint. I prioritize the backlog, I worked with the developers to break down the work, I checked the developer's work and did the acceptance criteria, I wrote the user stories, all those functions is what I see over and over and over again in product owner resume. What you want to do instead is say, "I led the," we can even do the login API,"I led the work around the login API. The problem that I was solving around it was trying to enhance security for our login protocols to meet regulatory requirements. I interviewed a bunch of users, I got up to speed on the regulatory requirements, I worked really closely with our legal teams and our compliance teams to translate that into something that was going to secure our bank, and when we launched, we were able to meet our compliance, save our bank a couple million dollars, and we smoothly transitioned into these new security requirements without disrupting any service for customers for four million customers."

(01:05:40):

Way different story than I prioritized backlog and I shipped it off to developers. Take it a step further. If you're working on customer-facing things, who are your customers? Did you go out and talk to them? By interfacing with customers and understanding them, I was able to solve XY, and Z problem with them, which resulted in a measurable amount of XY and Z metric going up for the business. I ran this function, I ran this feature, I launched this feature, I watched it through, I iterated on it, I did the stuff that was needed to make this successful. That's what I want to see on a resume.

(01:06:15):

Even if you have a product owner title, I'll still read the details and everybody else will too, but I will say there's sometimes a poor connotation when you have that title unfortunately because of the baggage that's associated with agile. Even just on resumes, I would say do product owner/product manager in there just to let people know that I know how to do this and I've been doing this well. If you do that in your bullet points, that shows up as well. There's also this whole concept that we didn't even get into about certifications. People keep asking me if I want to transition into product management, should I get a certification, an agile certification?" I feel like these were bigger a couple years ago, but they're still big. If you ever see somebody with a CSPO on the end of their profile, which you probably have seen on LinkedIn, it's a certified Scrum product owner.

(01:07:05):

Now, one thing to remember, and this is about all agile stuff, is we call it the agile industrial complex, agile coaches and agile trainers, the whole Scrum team, scrum.org, Scrum Inc, SAFe, everybody, the way they make their money is through consulting to teach you these processes and by having people be trained to get these certifications. They come in and they say, "All your people need to be certified Scrum product owners. Give me 2,500 bucks per person," and then they get a certificate at the end of a two-day class that says they're a certified Scrum product owner. It doesn't necessarily mean they could do the job, it doesn't necessarily mean they could do product management, but let's think about when we're talking about too should we adopt SAFe in general, should we adopt these things, think about how these organizations make money.

(01:07:56):

They're selling certifications. Of course they want more and more people to adopt it. That's the idea here. They're selling you this dream that you just certify all your people and then you could be working on it. They take all these people, they put them in two-day classes or whatever, and they turn them out and then they say, "Go, you're a completely new role." It doesn't work that way. That's not in the best interest of your company, that's not really what we're looking for here, and that's why all of this stuff needs to go deeper. If you've done a CSPO class and you have that certification, it may help you get hired at another large enterprise that is adopting Scrum and SAFe. That will probably help you there.

(01:08:36):

If you want to transition into tech and go into the companies that we talk about, they're probably going to look at that and say, "This person doesn't know what they're doing. This is not here." If you do know what you're doing and you did that for a reason, because some people need that to get promoted, some companies actually require it, which is crazy, but to get promoted or be that thing, I have tremendous sympathy for that, but you're going to have to do a lot of work explaining in your resumes and stuff as you transition that you know more than that. You're not just a CSPO with a two-day class, you have done the work. That's where all of this building up on your resume becomes really, really important. It's not just about getting certified.

(01:09:14):

I had people ask me, they're like, "Can you just certify product managers?" I'm like, "No. If people take my course, I give them a certificate of completion and as a completion." You finished a course just like any other course, but I will not certify product managers because I do not think you can ever say somebody is prepared and able to do their job from a short class. Now, there are some agile agencies that do a lot more training where you have different levels, and what they would do is they would train people, but then they would make them go do work and they had a coach that they worked with, and they go back and forth and they could demonstrate that they could do the work over time to get to the next level.

(01:09:56):

It's almost like the PMP, the project management certification where you have to have time actually doing the job. That's different, that's a different type of skill, it's a different type of certification, but if you see any CSPOs, it's typically a two-day workshop that they went to and then got certified. That's the difference with this. I would say be careful if you are a product owner wanting to be a product manager of just certifications. All the large tech organizations I know too, they're not looking for certifications in product management or product ownership to hire people, they're looking for experience, but the organizations that might not know what they're doing, they are looking for CSPOs, they are looking for that.

(01:10:37):

If it's required by your organization, you might have to ask, "Are we all well set up here to do our best job? Is this the place where I'm going to learn how to be a better product manager?" I also feel bad though for people because it's hard to be a product manager, it's hard to get your foot in the door. I'm so torn on it because there are organizations that hire people with a CSPO and they require it, so of course if it gets your foot in the door and it helps you do it, but if it's not going to help you and it's not going to put you on the career path you want, I don't think it's worth money.

中文翻译:

是的。当我看到 PO 写简历或描述工作职责时，他们总是从“流程”的角度切入。比如：“我负责待办事项的优先级排序”、“我与开发人员合作分解任务”、“我检查开发人员的工作并制定验收标准”、“我编写用户故事”。这些职责我在 PO 的简历中见过无数次。相反，你应该说：“我领导了（我们甚至可以用登录 API 做例子）登录 API 的相关工作。我解决的问题是尝试增强登录协议的安全性以满足监管要求。我访谈了一批用户，深入研究了监管要求，与法律和合规团队紧密合作，将其转化为保障银行安全的功能。发布后，我们达到了合规要求，为银行节省了数百万美元，并在不中断 400 万客户服务的情况下平稳过渡到了新的安全标准。”

(01:05:40):

这比“我排了优先级并交给开发”要动听得多。再进一步，如果你做的是面向客户的东西，你的客户是谁？你出去和他们谈了吗？通过与客户接触并理解他们，我解决了他们的 X、Y、Z 问题，从而使业务的 X、Y、Z 指标有了可衡量的提升。我负责这个功能，我发布了这个功能，我全程跟进，我进行了迭代，我做了让它成功所需的一切。这才是我想在简历上看到的。

(01:06:15):

即使你的头衔是 PO，我仍然会读细节，其他人也会。但我必须说，不幸的是，由于敏捷带来的包袱，这个头衔有时会带有负面暗示。即使在简历上，我也建议写成“产品负责人/产品经理”，以此告诉人们我知道怎么做，而且做得很好。如果你在要点（bullet points）中体现了这一点，它就会显现出来。还有一个我们还没聊到的

概念：认证。人们总问我：“如果我想转行做产品管理，我应该考个认证吗？敏捷认证？”我觉得这在几年前更火，但现在依然很有市场。如果你看到有人在 LinkedIn 个人资料后面加个 CSPO，那是“认证 Scrum 产品负责人”。

(01:07:05):

现在要记住一件事，关于所有敏捷的东西，我们称之为“敏捷工业复合物”（agile industrial complex）。敏捷教练、培训师、整个 Scrum 团队、scrum.org、Scrum Inc、SAFe，所有人，他们赚钱的方式就是通过咨询教你这些流程，以及让人们接受培训获得认证。他们进来说：“你所有的人都需要成为认证 Scrum 产品负责人。每人给我 2500 美元。”然后人们在两天的课程结束后拿到一张证书。这并不一定意味着他们能胜任工作，也不意味着他们懂产品管理。但想想看，当我们讨论是否应该采用 SAFe 时，想想这些组织是怎么赚钱的。

(01:07:56):

他们在卖认证。当然，他们希望越来越多的人采用它。这就是核心逻辑：他们卖给你一个梦想，只要认证了所有人，你就能搞定。他们把人塞进两天的课程，然后把他们吐出来说：“去吧，你现在是一个全新的角色了。”但现实并非如此。这并不符合你公司的最佳利益，也不是我们真正追求的东西。这就是为什么这些东西需要更深入。如果你上过 CSPO 课程并拿到了认证，它可能会帮你被另一家正在采用 Scrum 和 SAFe 的大型企业录用。在那儿可能有用。

(01:08:36):

但如果你想转行进入我们讨论的那些科技公司，他们可能会看着认证说：“这个人不知道自己在做什么。”如果你确实懂行，而且考证是有原因的（比如有些人需要它来晋升，有些公司竟然强制要求，这很疯狂），我非常同情。但在你转型时，你必须在简历中做大量解释工作，证明你懂的不止这些。你不仅仅是一个上了两天课的 CSPO，你真正做过实战。这就是为什么在简历中积累实战经验变得极其重要。这不仅仅是关于考证。

(01:09:14):

有人问我：“你能给产品经理发认证吗？”我说：“不。如果人们上了我的课，我给他们结业证书。”你完成了一门课程，仅此而已。但我不会给产品经理发“认证”，因为我不认为你能通过一门短期的课就断定某人已经准备好并有能力胜任工作。现在有一些敏捷机构做更深入的培训，分不同等级。他们会培训人，但要求他们去实战，并配有教练，通过长时间的反复实践来证明他们有能力进入下一阶段。

(01:09:56):

这有点像 PMP（项目管理专业人士认证），你必须有实际从事该工作的时长。那是另一种技能，另一种认证。但如果你看到 CSPO，那通常只是一个两天的研讨会。这就是区别。我会说，如果你是一名想成为 PM 的 PO，要警惕“唯认证论”。我认识的所有大型科技组织在招聘时都不看产品管理或 PO 的认证，他们看重的是经验。但那些可能不太懂行的组织，他们确实在找 CSPO。

(01:10:37):

如果你的组织有要求，你可能得问问：“我们在这里是否具备了做好工作的条件？这里是我学习成为更好 PM 的地方吗？”我也为这些人感到难过，因为做 PM 很难，入行也很难。我很纠结，因为确实有组织要求 CSPO 才能入行，如果它能帮你敲开大门，那当然好。但如果它不能帮你走上你想要的职业路径，我认为它不值那个钱。

[01:11:14] Lenny Rachitsky

English:

I think one interesting thread throughout this whole conversation is rarely is a plug-in play-ish easy solution going to be the answer to your problem, whether it's plugging in SAFe and it's going to help us build great software, taking a class helping you become a great product manager, be skeptical of that.

中文翻译:

我认为整个对话中一个有趣的线索是：很少有那种“即插即用”的简单方案能解决你的问题。无论是引入 SAFe 就能帮我们构建伟大软件，还是上一门课就能让你成为优秀 PM，都要保持怀疑。

[01:11:31] Melissa Perri

English:

Yeah. There's no quick way to doing any of this, there's no fast track. You don't get to skip over all the hard things.

中文翻译:

是的。做这些事没有捷径，没有快车道。你无法跳过所有那些困难的部分。

[01:11:39] Lenny Rachitsky

English:

Yeah. Bummer.

中文翻译:

唉，真遗憾。

[01:11:41] Melissa Perri

English:

Yeah. I wish.

中文翻译:

是啊，我也希望有。

[01:11:43] Lenny Rachitsky

English:

Yeah. One question I wanted to clarify. When you come into an org and they have product owners, do you encourage them to get rid of product owner as a title and role and make them product managers or do you keep product owners?

中文翻译:

对。我想澄清一个问题。当你进入一个拥有 PO 的组织时，你会鼓励他们取消 PO 这个头衔和角色并把他们变成 PM 吗？还是保留 PO？

[01:11:53] Melissa Perri

English:

I say that we should have a hierarchy. You would have all product managers on a team, they would be an IC, individual contributor, so they're either an associate product manager and that's if they don't have all the discovery experience or maybe they know basic Scrum, totally fine, you can be an associate product manager, but if they don't know how to talk to customers, digest what the customers are saying and turn that into a feature direction and a backlog and this is how we're going to work, all that stuff needs to be in there. If they don't know how they should be measuring things, you're not quite a product manager yet.

(01:12:28):

Associate product manager, and then product manager, and then a senior PM. A senior PM can work on a Scrum team as well or a development team. I do encourage them, I say, "Get rid of these two different titles because it's confusing everybody." I help a lot of people with career paths too. I've seen companies with 47 different titles for product managers because somebody in this organization is called a product associate and somebody over here is a product manager and that person is a platform product manager, and that person is a platform product owner, and this person is an API product owner. You can have product manager, senior product manager, all that stuff, but I would not confuse people with the two different titles of product owner versus product manager.

中文翻译:

我说我们应该建立一个层级体系。团队中应该全是产品经理，他们是 IC（个人贡献者）。如果他们没有足够的探索经验，或者只懂基础的 Scrum，那可以是“助理产品经理”（Associate PM），这完全没问题。但如果他们不知道如何与客户交谈、消化客户反馈并将其转化为功能方向和待办事项，以及确定工作模式，那么这些东西都必须补齐。如果他们不知道如何衡量事物，那他们还不算是一个真正的产品经理。

(01:12:28):

层级应该是：助理产品经理，然后是产品经理，然后是高级 PM。高级 PM 也可以在 Scrum 团队或开发团队中工作。我确实鼓励他们：“取消这两个不同的头衔，因为这让每个人都感到困惑。”我也帮很多人制定职业路径。我见过有的公司有 47 个不同的产品经理头衔，因为这边叫产品助理，那边叫产品经理，那个人叫平台产品经理，那个人叫平台产品负责人，这个人叫 API 产品负责人。你可以有产品经理、高级产品经理等等，但我不会用“产品负责人”和“产品经理”这两个不同的头衔来迷惑大家。

[01:13:11] Lenny Rachitsky

English:

But importantly, there's a bar for who is called a product manager because if you take a product owner as you've said, and say you're a product manager, they're going to be a terrible product manager. Your advice is make them an associate PM, and then once they reach a certain level, they've graduated to product manager.

中文翻译:

但重要的是，对于谁能被称为“产品经理”是有门槛的。因为正如你所说，如果你直接把一个 PO 叫作 PM，他们可能会是一个糟糕的 PM。你的建议是先让他们做助理 PM，等达到一定水平后，再“毕业”成为产品经理。

[01:13:28] Melissa Perri

English:

It depends. I don't say this as a hard and fast rule because there are product owners out there who've been doing their job very well. Just because they have a title of a product owner doesn't mean that somebody can't do the job of a product manager. There are plenty of people out there who were just named this because a company names them that and they know what they're doing, so don't look at that hard and fast. When we do this, we typically will say everybody's got product owner on their title, change it to product manager, but we will go back and look at what is the actual skillset.

(01:13:56):

When I've come in to work with companies on transformations, what we typically do, the way I get introduced usually is we come in and they're looking for some kind of training for all of their product teams, and then we bring in Product Institute and we do our online training, Then afterwards I work with the organization, I say, "Now that everybody's been baselined and trained, we have to figure out who is going to be a great product manager and who's not," so there's an awareness. You know what's interesting? Once people figure out what the job's about, a lot of them opt out.

(01:14:24):

When I did the transformation at Athenahealth, we had 365 product managers and they all had different titles too, product innovation, all over the place. We turned them into product managers, we trained everybody, we gave everybody the opportunity to learn, which I think is great, and then we gave them opportunity to practice their skills. What happened is at the end of the training, a lot of people raised their hand and said, "Oh, no, no, this is not what I wanted to do. I thought it was very different. I did not realize how much people work it was. I have to go influence these people, I have to do all this stuff. This is actually not really what I wanted to do."

(01:14:59):

We found other roles for them that were what they wanted to do or they decided to leave. That was up to them, we didn't cut anybody, but we moved some people into operations because they wanted to be a little more heads down to find process. That was their thing. We moved people into data roles who were good with SQL and were good with data analysis. We moved people into user research roles because they wanted to talk to customers, but they didn't want the responsibility and the accountability that came with the product management piece because they realized it was so intense.

(01:15:29):

I've seen this happen a lot in large organizations. You baseline everybody, you show them what the role is and then you let them go practice it, and then at that point some people will opt out and then you have to go back through your people and say, "Okay. How do we level-set now? Who's doing really well? Who's not doing so well? Which teams need to have more experienced people on it because they don't have anybody to learn from?" That's where we would say, "Hey, let's hire some ICs over here or a director of product management who can help train these people and help them keep growing." That's been super successful. I've seen people bring in some more directors, scattered them around, and they've leveled up these product owners who were not necessarily doing great product management and now they're doing fantastic.

(01:16:13):

I've watched fantastic leaders in these organizations that are not software native or doing these transformations. Bringing the right person, you can make amazing product managers. Give them a year or two and completely turn it around. It's totally possible. It's totally possible to take people and train them, and I firmly believe in that, but you got to get them exposure to what good looks like. If you are in an organization and you cannot see what good looks like anywhere, that's a red flag. That's where leaders need to look at their organization and say, "Do I have people with these skills interspersed around so that everybody can start to learn, so that everybody can actually be on the uptake of this and make sure that we are well-balanced?"

中文翻译:

这取决于具体情况。我不会把它当作死板的规定，因为确实有些 PO 做得非常好。仅仅因为头衔是 PO，并不意味着某人不能胜任 PM 的工作。有很多人只是因为公司这么命名才叫这个头衔，但他们其实很懂行，所以不要

死扣头衔。当我们做转型时，通常会说：把每个人的头衔都改成产品经理，但我们会回头审视实际的技能组合。

(01:13:56):

当我进入公司协助转型时，通常的流程是：他们寻求对所有产品团队进行某种培训，然后我们引入 Product Institute 进行在线培训。之后，我会与组织合作说："现在每个人都经过了基准培训，我们得弄清楚谁会成为优秀的 PM，谁不会。"这样就有了认知。你知道有趣的是什么吗？一旦人们弄清楚这份工作到底是干什么的，很多人会选择退出。

(01:14:24):

当我在 Athenahealth 做转型时，我们有 365 名产品经理，头衔也是五花八门，什么"产品创新"之类的。我们将他们统一为 PM，培训了每个人，给了每个人学习的机会（我觉得这很棒），然后给他们实践技能的机会。结果是，培训结束时，很多人举手说："噢，不，不，这不是我想做的。我以为它很不一样。我没意识到这需要做这么多'人的工作'。我得去影响这些人，我得做所有这些事。这其实不是我想要的。"

(01:14:59):

我们为他们找到了其他符合意愿的角色，或者他们决定离开。这取决于他们，我们没有裁掉任何人。我们将一些人转到了运营岗位，因为他们想更专注地研究流程，那是他们的强项。我们将擅长 SQL 和数据分析的人转到了数据岗位。我们将想和客户谈话但不想承担 PM 那种责任和问责制的人转到了用户研究岗位，因为他们意识到 PM 的工作强度太大了。

(01:15:29):

我在大组织中经常看到这种情况。你给每个人设定基准，展示角色内涵，让他们去实践。到那时，有些人会退出，然后你必须重新审视你的团队："好吧，我们现在如何重新对齐？谁做得很好？谁做得不太好？哪些团队需要更有经验的人，因为他们在那儿没人可学？"这时我们会说："嘿，让我们在这里招一些 IC，或者招一个产品管理总监来帮助培训这些人并让他们持续成长。"这非常成功。我见过人们引入更多的总监，分散到各个部门，他们提升了那些原本不一定擅长产品管理的 PO，现在他们做得棒极了。

(01:16:13):

我在这些非原生软件公司或正在转型的组织中见过非常出色的领导者。只要引入正确的人，你就能培养出惊人的产品经理。给他们一两年的时间，就能彻底扭转局面。这完全是可能的。我坚信可以选拔人才并进行培训，但你必须让他们接触到"什么是优秀的"。如果你在一个组织里，到处都看不到"优秀"长什么样，那就是一个危险信号。这时领导者需要审视组织并问："我是否在各处穿插了具备这些技能的人，以便每个人都能开始学习，确保每个人都能跟上进度并保持平衡？"

[01:16:56] Lenny Rachitsky

English:

That was an awesome nuance and addition. I'm just looking back at all the things we've talked about. We've covered so much ground, the history of agile and Scrum and SAFes and product owners, we've gotten into all kinds of advice on how to do this better as a company leader, as a product owner, as someone thinking about even moving through product management. Is there anything else that we have not covered that you think might be helpful to touch on or wrap up on?

中文翻译:

这个补充非常精彩且细致。回顾一下我们聊过的内容，我们涵盖了非常广的领域：敏捷、Scrum、SAFe 和 PO 的历史，还有给公司领导者、PO 以及想转行做产品管理的人的各种建议。还有什么我们没聊到、你觉得会有帮助或适合作为总结的内容吗？

[01:17:25] Melissa Perri

English:

I think I would tell people to remember that when you look at agile methodologies, and if you look at it with small agile, what we are really saying there is we want to be able to move quickly and deliver great value to customers. If you embrace those principles, you're going to do well, but if you think of agile as just a defined super cut and dry process where you have to follow every single little step here and there, that's not going to serve you because you're not getting back into why are we doing this.

(01:17:54):

There are some great agile coaches out there. There are first principle approaches to doing great product work and doing great development work. They're there because they are not just selling Scrum, they are there to make people better. Those agile coaches I think can make great fantastic teams, and I've worked with a lot of them, and I think they are really there to just make it a better environment for the people who are working and to help the company produce better products. That's their whole goal.

(01:18:21):

Then there are people too who are wedded to this is SAFe and you will do it by the book. This is Scrum, you will do it by the book. I'd be very skeptical of that because what's the end goal? What's their end goal? Like I said, a lot of people out there get paid by certifying people and by consulting on these processes. McKinsey made a huge division to do this, and a lot of SAFe was actually introduced to organizations from McKinsey and from large organizations, consulting organizations like that. They came in and they said, "Yeah, this is what you do. I'll teach you how to do SAFe, I'll teach you how to do Scrum." They have been building those consulting agencies off of agile transformations.

(01:19:03):

We always laugh when I talk to other people who've been doing transformations and coaches. I go, "Why is McKinsey always coming in here and doing this?" I've followed them into so many organizations and I've been like, "Oh, wow, they screwed this up. Let me go fix it." There's always a saying that nobody gets fired for hiring McKinsey. McKinsey's big, people trust their name. A lot of the people at McKinsey haven't done this before, they've never been inside the organization trying to transform it and stay there for a long time. That's why I'd be really skeptical of who's selling that.

(01:19:37):

If you approach agile from a perspective of I want to be agile because I want to release things quickly and get feedback from customers and make sure that that's great, look at all of your processes like that and say, "Is that serving the best interest of our company and our culture and our customers? Is this making our customers proud of us? Is this helping our customers receive good value?" If your processes aren't, fix them, change them, inspect and adapt, ease a lowercase agile principle. We should be looking at every process we do and saying, "Is it working?", and not be afraid to change it.

中文翻译:

我想提醒大家记住，当你审视敏捷方法论时（如果是指“小写”的敏捷精神），我们真正的意思是：我们希望能够快速行动并为客户交付巨大价值。如果你拥抱这些原则，你会做得很好。但如果你把敏捷看作是一套定义死板、枯燥乏味的流程，必须一步步照做，那对你没好处，因为你没有回到“我们为什么要这样做”的原点。

(01:17:54):

外面确实有一些很棒的敏捷教练。他们采用“第一性原理”的方法来做优秀的产品工作和开发工作。他们存在不是为了推销 Scrum，而是为了让人变得更好。我认为这些教练能带出极棒的团队，我与其中很多人合作过。他们的目标只是为员工创造更好的环境，并帮助公司生产更好的产品。这就是他们的全部目标。

(01:18:21):

但也有些人死守着“这就是 SAFe，你必须照本宣科”或者“这就是 Scrum，你必须照做”。我会对此保持高度怀疑，因为终极目标是什么？他们的目标是什么？正如我所说，很多人是靠给人发证和做流程咨询赚钱的。麦肯锡专门成立了一个庞大的部门来做这个，很多组织的 SAFe 其实是由麦肯锡或类似的大型咨询机构引入的。他们进来说：“是的，你就该这么做。我教你做 SAFe，教你做 Scrum。”他们靠着“敏捷转型”建立起了这些咨询业务。

(01:19:03):

当我与其他做转型的人和教练聊天时，我们总是笑。我会问：“为什么麦肯锡总是跑进来干这个？”我跟在他们后面进过很多组织，心想：“噢，哇，他们把这儿搞砸了，让我来修补吧。”俗话说“没人会因为雇佣麦肯锡而被开除”。麦肯锡名气大，人们信任这个名字。但麦肯锡的很多顾问以前没做过这个，他们从未真正待在组织内部尝试长期转型。这就是为什么我会对推销这些东西的人保持警惕。

(01:19:37):

如果你从“我想变得敏捷，因为我想快速发布并获得客户反馈，确保产品出色”的角度出发，那么请这样审视你所有的流程：“这是否符合我们公司、文化和客户的最佳利益？这是否让我们的客户为我们感到自豪？这是否帮助客户获得了良好的价值？”如果你的流程做不到，那就修复它、改变它、检查并调整——这才是“小写”的敏捷原则。我们应该审视我们做的每一个流程并问：“它有效吗？”，并且不要害怕去改变它。

[01:20:15] Lenny Rachitsky

English:

I think it's important to highlight this can work. There are ways to reorganize the way you build product at a very large company where you can actually deliver great product consistently. You've seen that happen a lot. Anything you want to add there to give people hope?

中文翻译:

我认为强调“这是行得通的”很重要。在大公司里，确实有办法重新组织构建产品的方式，从而能够持续交付出色的产品。你见过很多这样的案例。关于这一点，你有什么想补充的来给大家一些希望吗？

[01:20:30] Melissa Perri

English:

Yeah. The biggest pushback I hear from large organizations, especially ones that are not software native as banks or insurance companies, whatever it is, is that we're not like Google, we're not like SaaS companies. It's true, you're not a SaaS company. The way that you do it is going to be slightly different than the way that Amazon runs or the way that Google runs. I don't see many companies actually run the same, and just because it works at Google doesn't mean it's going to work at an insurance company that's a hundred years old. That's okay, but you could still learn from people who've been producing software at scale for many, many years. You can still learn what makes them successful, and then you could take some of those principles and apply them to your organization and then figure out where we need to adapt. That's what I would look at.

(01:21:21):

I see software strategies and digital strategies becoming so intertwined into the strategies of companies in general, whether you're an insurance company or a bank or anything. They're becoming so heavily entwined in the C-suite strategies. The first thing I would say, and where I see a lot of companies struggle with this is you have to make sure that it is thought of at the C-suite. What some organizations do is go, "Oh, no, that's IT work. Let me push all the software strategy down." If you're doing that, you're missing out on innovation, and this is where the product role comes in and where I think more organizations need chief product officers because they're the person in the C-suite saying, "How can software enable us to be 10 times better and crush our competition, whether we're a bank or insurance company or a pharmaceutical company? What can we do with our software that's going to put us ahead of the competition, that's going to put us ahead of the market?"

(01:22:16):

If you're not having those conversations when you're thinking of your long-term company strategy, you're behind, you're behind because that is what every small high-growth startup is doing that's coming out to try to disrupt these big companies. The big companies are successful, they have a lot of money so they don't have as much urgency as sometimes these smaller companies do to change or somebody who's failing to change. That's why things are going so slow, but at the same time, if we don't pay attention to that and we're not considering it, that's where the danger comes in.

中文翻译:

是的。我从大组织（尤其是银行、保险公司等非原生软件公司）听到的最大阻力是："我们不像谷歌，我们不像 SaaS 公司。"没错，你确实不是 SaaS 公司。你做事的方式会与亚马逊或谷歌略有不同。我没见过哪两家公司的运作方式是完全一样的。仅仅因为某种方式在谷歌奏效，并不意味着它在一家百年历史的保险公司也奏效。这没关系。但你仍然可以向那些多年来大规模生产软件的人学习。你仍然可以学习是什么让他们成功，然后提取其中的一些原则应用到你的组织中，再弄清楚哪里需要调整。这就是我关注的点。

(01:21:21):

我看到软件战略和数字化战略正与公司的整体战略紧密交织，无论你是保险公司还是银行。它们正深度融入高管层的战略中。我要说的第一点（也是我看到很多公司挣扎的地方）是：你必须确保高管层考虑到了这一点。有些组织的做法是："噢，不，那是 IT 的活儿，把所有软件战略都往下推。"如果你这样做，你就错失了创新。这就是产品角色的切入点，也是为什么我认为更多组织需要首席产品官（CPO）的原因。因为 CPO 是高管层中那个会问"软件如何让我们变得强 10 倍并击败竞争对手"的人。无论我们是银行、保险还是制药公司，我们的软件能做什么来让我们领先于竞争对手、领先于市场?

(01:22:16):

如果你在思考公司长期战略时没有进行这些对话，你就落后了。你落后是因为每一个试图颠覆大公司的、高增长的小型初创公司都在这样做。大公司很成功，很有钱，所以他们有时不像小公司或面临失败的公司那样有紧迫感去改变。这就是为什么进展缓慢的原因。但与此同时，如果我们不关注这一点，不考虑它，危险就会随之而来。

[01:22:46] Lenny Rachitsky

English:

I imagine people are going to have a lot of questions and hope to get even more advice. A couple of final questions. Where can folks find you online if they want to ask more questions along these lines, maybe get your help on some of this stuff? Then finally, how can listeners be useful to you, Melissa.

中文翻译:

我想大家肯定会有很多问题，并希望能得到更多建议。最后两个问题：如果大家想就这些方面提问，或者想寻求你的帮助，可以在哪里找到你？最后，听众们能为你做些什么?

[01:23:02] Melissa Perri

English:

You can find me on LinkedIn. If you go search for Melissa Perri with an I, P-E-R-R-I, you'll find me there. My LinkedIn is /melissajeanPerri. I'm happy to connect with people there. My school is called Product Institute, so if you're interested in doing product management training or getting help with some of this stuff, go to productinstitute.com and reach out to me. I run a podcast too called The Product Thinking Podcast. We answer questions every week, and a lot of them are around SAFe, agile and all these topics, so if you have a question and you're saying, "Hey, how do I do this?", definitely reach out and let me know.

中文翻译:

你可以在 LinkedIn 上找到我。搜索 Melissa Perri（末尾是 I，P-E-R-R-I）就能找到。我的 LinkedIn 账号是 /melissajeanPerri，我很乐意在那里与大家交流。我的学校叫 Product Institute，所以如果你对产品管理培训感兴趣，或者想在这些方面寻求帮助，请访问 productinstitute.com 并联系我。我也主持一个播客叫《产品思维播客》（The Product Thinking Podcast）。我们每周都会回答问题，其中很多是关于 SAFe、敏捷和这些话题的。所以如果你有问题想问“嘿，我该怎么做？”，一定要联系我。

[01:23:35] Lenny Rachitsky

English:

Awesome. Melissa, I learned a lot. I feel like a lot of people who listen to this podcast, they're like, "I had no idea about any of this, and now I understand all these terms that I kind of hear occasionally." Thank you for coming and sharing all of this with us, especially the advice for folks that are in this world right now.

中文翻译:

太棒了。Melissa，我学到了很多。我觉得很多听这个播客的人会想：“我以前对这些一无所知，现在我终于理解了那些偶尔听到的术语。”感谢你过来与我们分享这一切，特别是给那些目前正处于这个领域的人们的建议。

[01:23:49] Melissa Perri

English:

Yeah. Thanks for having me.

中文翻译:

是的，感谢邀请我。

[01:23:51] Lenny Rachitsky

English:

Bye everyone.

中文翻译:

大家再见。

[01:23:52] Melissa Perri

English:

Bye.

中文翻译:

再见。

[01:23:55] Lenny Rachitsky

English:

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:

非常感谢您的收听。如果您觉得本节目有价值，可以在 Apple Podcasts、Spotify 或您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本播客。您可以在 lennyspodcast.com 找到所有往期节目或了解更多信息。下期节目再见。