The way we think about product development at Stripe,
it really is to find the correct set of early users
to kind of co-create the product with.
Maybe the best example of that is Stripe Billing.
When we got to starting the Stripe Billing product,
we realized that there were a number of our existing users.
These were companies like Figma and Slack,
and we were already using Stripe for payments
that had these subscriptions business models,
and we figured that there were gonna be many more
of these kind of companies into the future,
and we could see that they were really kind of pushing
the boundaries of what was possible here.
So we decided to co-create the product with them.
So we had shared Slack channels,
we'd actually show them product on a very regular basis,
get their feedback on it.
And only when that original kind of alpha group
was super, super happy with the products
did we then think it might be ready
to go to a broader audience.
So that is just how we build product at Stripe,
and that means that every engineer
building product at Stripe really has many of the kind
of attributes and will exercise many of the attributes
that you'll often find in PMs and in other companies.
Welcome to Lenny's podcast,
where I interview world-class product leaders
and growth experts to learn from their hard-won experiences
building and growing today's most successful products.
Today, my guest is David Singleton.
David is chief technology officer at Stripe,
where he's responsible for guiding its engineering
and design teams, a role he's had for over five years.
Prior to Stripe, David was VP of engineering at Google,
where he spent over a decade,
and in hearing from David,
you'll quickly be able to tell how passionate he is
about the craft of building great products
and building great teams.
We dig into Stripe's unique approach to hiring,
how they built a very product-oriented engineering team,
which allowed them to hold off on hiring

their first product manager for many, many years,
how they operationalize their operating principle
of be meticulous in your craft,
including a number of fascinating internal processes
like engineer occasions, walking the store,
and something called friction logging.
David also shares what it takes to run an engineering culture
with the uptime and scale requirements of a Stripe,
plus how AI is already impacting how they work
and lessons around leadership, management, and planning.
I am so thankful David carved out time to come on the podcast,
and I know that you'll learn something valuable from it.
With that, I bring you David Singleton
after a short word from our sponsors.
This episode is brought to you by Mixpanel,
offering powerful self-serve product analytics.
If you listen to this podcast,
you know that it's really hard to build great product
without making compromises.
And when it comes to using data,
a lot of teams think that they only have two choices,
make quick decisions based on gut feelings,
or make data-driven decisions at a snail's pace.
But that's a false choice.
You shouldn't have to compromise on speed
to get product answers that you can trust.
With Mixpanel, there are no trade-offs.
Get deep insights at the speed of thought
at a fair price that scales as you grow.
Mixpanel builds powerful and intuitive product analytics
that everyone can trust, use, and afford.
Explore plans for teams of every size
and see what Mixpanel can do for you at mixpanel.com.
And while you're at it, they're hiring.
Check out mixpanel.com to learn more.
This episode is brought to you by Epo.
Epo is a next-generation AB testing platform
built by Airbnb alums for modern growth teams.
Companies like DraftKings, Zapier, ClickUp, Twitch,
and Cameo rely on Epo to power their experiments.
Wherever you work, running experiments
is increasingly essential,
but there are no commercial tools

that integrate with a modern growth team stack.
This leads to waste of time building internal tools
or trying to run your own experiments
through a clunky marketing tool.
When I was at Airbnb,
one of the things that I loved most about working there
was our experimentation platform,
where I was able to slice and dice data
by device types, country, user stage.
Epo does all that and more delivering results quickly,
avoiding annoying prolonged analytic cycles
and helping you easily get to the root cause
of any issue you discover.
Epo lets you go beyond basic quick-through metrics
and instead use your NorthStar metrics
like activation, retention, subscription, and payments.
Epo supports tests on the front end, on the back end,
email marketing, even machine learning clients.
Check out Epo at geteppo.com.
That's geteppo.com and 10x your experiment velocity.
David, welcome to the podcast.
Thanks for having me, Lenny.
It's great to be here.
I know Stripe Sessions is coming around the corner
in later next week,
and I imagine it's a pretty hectic time for you.
And so I just wanna extra appreciate you making time
for this in the middle of all that.
Thank you.
Well, it's a pleasure.
It's a lot of fun.
Stripe Sessions is our annual user conference.
So we get together a bunch of folks
from businesses building on Stripe.
And it's a great opportunity both to share
what we've been up to over the last year,
but also just learn a lot from them about, you know,
what they would like to see us doing.
And so I find it very energizing.
It is, of course, a lot of work to put it together.
In particular, I'm spending a bunch of time
getting some demos together today,
which I love building demos.

So, but it's a real pleasure to join you today.
So as I was preparing for this conversation,
I kind of realized that the Stripe alumni
are just killing it on this podcast,
both in terms of the number of people
that have come from Stripe,
and also just in terms of the quality
and the popularity of some of the episodes.
So folks like Claire Hughes-Johnson
and Shreyas Doshi and Aika.
And my first question essentially,
what is it that you all do that allows you to attract,
hire, close, keep people of that caliber?
And I'm not gonna accept just we keep a high bar
or we just spent a lot of time on hiring.
I'm curious just like, what is it that you all do
that you think that other companies don't do
that's maybe unique to the way Stripe finds and hires?
Well, yeah, I'd love to tell you more
about how we find and hire people.
I think it makes sense to kind of zoom out for a second
because it's material here.
So think about what Stripe's doing.
Our core thesis is that the internet economy
is going to be much bigger and more important
in the future than it is today.
And that certainly played out over the last 10 years.
And so we are really a business full
of product-minded builders who are seeking
to make it easier for businesses to get started
and to operate against that backdrop.
And so we got started building a credit card payments API
before Stripe came along,
accepting credit card payments online was way too hard.
You had to go talk to your bank.
You had to maybe sign a contract with Visa MasterCard.
You then had to kind of string together
all this very complicated infrastructure
and then put a bunch of kind of restrictions
and requirements on the business.
And John and Patrick are co-founders
at a previous business and they tell this funny story
actually, which is while they were building that company,

they assumed it was an internet company.
They assumed the hardest thing was going to be
like putting together hardware in a data center,
but it turned out because there were services like AWS
around by then that that was actually quite easy.
However, when it came to accepting payments
in that business, it was just as hard
as they had expected building hardware to be.
They had to actually deploy some physical hardware.
They had to do a whole bunch of these partnerships.
And it was very difficult, especially as an upstart,
folks that didn't have any reputation yet
to kind of be taken seriously
and be able to correct into that.
And so that's kind of the inspiration behind Stripe.
So we started out solving a really kind of important
and profound problem for these businesses
getting started and scaling
and obviously taking a very developer-centric approach
to that in the early days.
I think what's exciting to know about
and important to know about next is as we've scaled,
we really kind of figure out what to do
based on paying really close attention to our users.
Those are the businesses building on Stripe.
What are they getting out of the products as they are today?
And what are the problems adjacent
to the ones that we're already solving
that they have to put a lot of their own energy into
but really feel like they shouldn't
and they'd like us to help?
And when we find that confluence of kind of need
and then the adjacency to what we're already doing,
that's usually our invitation
to go and build the next layer
of our economic infrastructure.
So we think what we do today
is building economic infrastructure
for the internet and beyond.
So to bring this back to like,
how do we attract the right talent
and what does this matter?
First of all, a lot of people,

most people I think come to Stripe
because they see that there is this great opportunity
to help businesses at scale.
So it really is about taking part in a mission that matters
and are really motivated by that idea
that we spend a lot of time with our users,
the folks that actually need our help
and are guided by them.
And that mindset, I think,
attracts a certain kind of person
who wants to take a lot of agency,
like wants to understand the problems deeply
and come up with their own ideas
about how to solve for them.
Because we're building infrastructure,
it's also our approach that we really do think
that we can do the best work kind of behind the scenes.
We don't have to be having the splashy launch,
but hopefully all of our users can
and we are quietly working behind the scenes
to make their businesses better and more effective over time.
And I think that attracts folks
with a certain kind of mindset,
which is the interest and kind of tenacity
to dig into kind of difficult problems
that may not be obvious
and obviously kind of attractive to everyone in the world,
then really plug away at them
until we've made a tremendous amount of progress.
And so it attracts folks that are long-term thinkers,
that care about building
and also actually want to work very collaboratively together.
So when I think about what makes Stripe different
to other companies,
it really is this singularity of purpose
or mission is increasing the GDP of the internet
and building this financial infrastructure to do that.
And a tremendous amount of teamwork
and collaboration to get that done.
And so that attracts a certain kind of person.
When we think about finding those people,
we don't hide any of that.
Like you have to care about that

in order to, I think, be excited about working at Stripe.
And so we work very hard to make it very clear
as we're bringing people in that that's what this is about.
And to that end,
we're often very patient when we're hiring.
So especially for critical roles,
leadership hiring in particular,
we're happy to take our time
and meet tens or hundreds of people.
And then in particular,
sometimes we'll meet someone
that we think is a really good fit for a given role,
but they're not available right now.
And then we'll be patient.
We'll keep building the personal relationship with them.
And oftentimes the moment of time comes
when it makes sense for them to join.
And in order to pull that off,
it's also the case that hiring at Stripe
is a very personal activity.
So look, there are lots of other companies
where recruiting is kind of like this machine.
And by the way, we have an awesome recruiting team,
but folks across the rest of the company
partner very closely with our recruiting team.
So it's not like it's a machine that delivers new hires,
and then you're like,
oh, maybe this person is here
and I can put them to use them that thing.
Our managers spend a lot of time
identifying exactly what they need
for the various roles that we have.
And then getting to know the potential candidates,
the people out there in the world that could do these jobs,
actually like really go quite deep with them
on why it might make sense for them to feel like,
number one, they can like feel very personally fulfilled
against that mission here.
And then also why it represents like a bigger opportunity
for them than maybe anything else
they can spend their time and attention on.
And to that end,
another thing that I think Stripe has definitely delivered

for many, many people who join here is just,
and I care about preserving this in the culture,
is just tremendous learning opportunities.
Very, very many people at Stripe today
are doing a quite different job
to the one that they joined for.
In order to do that,
they've often had to learn a lot
and stretch themselves into new areas.
It's also the case that,
you think about that financial infrastructure we're building,
the space really rewards people
who want to go and learn a ton of detail
in how these systems that are like,
not that popular or well understood in the broader world,
how they actually work
and assimilate just a lot of context.
And then in particular,
you can build better stuff at Stripe
and you're sharing that context with each other
and helping other people learn.
So I think that all ladders up
to this kind of group of folks that you mentioned
and all the folks you mentioned
are wonderful Stripes or former Stripes
that come together in service for users.
I always love to hear what people call their people internally.
So Stripes, good to know.
That's right.
So just to summarize what you shared,
which I was just writing as you were talking about,
the things that essentially help Stripe
build this incredible team,
a strong mission and a mission that draws people in
that are incredible.
And people may be hearing them be like,
yeah, yeah, like sure.
But in my experience,
having invested in a bunch of companies
and working with a lot of founders,
there's such a dichotomy between how hard it is
for companies to hire when they don't have a mission.
People really care about it versus a mission

that's like really meaningful.
So there's something so powerful there.
And so it's 100%.
I could see how that works so well.
And then the other things you mentioned,
patience, personal connections
between the hiring manager and the people they're hiring.
And I see that on Twitter.
A lot of PMs are just like doing things.
Like basically it's like they're writing
on a little startup on Twitter,
like Jeff Weinstein.
Yeah, so Jeff runs, Jeff is the PM for Atlas product.
And he's a great example of someone
who cares a lot about learning exactly how the world works.
To me, for instance, one of the things
that that team and Jeff really has driven this himself
have done recently is recognizing
that when you're starting in your business in the US,
one of the most frustrating steps
is that you have to apply
for this employer identification number from the IRS.
And it used to have like an extremely long backlog.
And it's really hard.
You can't like start running your business
until you have it.
And Jeff dug into the details,
the team dug into the details.
And we're just like kept asking,
does it have to be this way?
Does it have to be this way?
And eventually we work with the IRS
to make it possible for us to issue those numbers
much, much more quickly, like instantly as you sign up.
And I think that's exactly a great example
of how being curious, having this relentless passion
to solve problems for users,
and then just keep plugging away to problem
delivers this great result and you know,
an environment where you can learn not from each other.
I hope he got a big promotion for getting the IRS
to do something that he needed.
That's incredible.

I'm gonna try to go one layer deeper on the hiring piece
and then move on to a different topic,
but what is just the interview process like?
I don't know how involved you are
with the product management hiring process,
but just I'm curious what that is like high level
and an engineering hiring process, however you can share.
Yeah, there are a couple of things that are very common
across hiring for all rules at Stripe
and in particular engineers and PMs.
One of the things is we do have structured hiring loops.
So we put everyone through a very consistent process.
So very personal as we help folks figure out
why they might wanna be here,
but then we have a consistent way of evaluating talent.
And I think that's important because it means
that you can as an interviewer, as a hiring manager,
you can actually really start to understand
what is the broad spectrum of responses you might get here
and what do I actually learn about individual candidates
when I'm asking like similar questions.
The other thing that all these loops have in common is
nothing is a trick question.
Like we try to put people into an environment
which is as close to doing the actual work together
as you possibly can.
So for engineers, for instance,
we've got several different coding exercises that they do.
They share their screen with the interviewer.
It's actually kind of pair programming.
You're welcome to like use Google to search for Stack Overflow
to search for answers.
We don't care, like we just wanna see the outcome.
You're happy to, we're happy for you
to ask your interviewer questions as you go along.
And we've created these exercises
that are as close to real problems
as we need to solve as much as possible.
The same is true on the PM side.
One of the things, for instance,
we do with PMs is we have a written exercise.
We've got obviously a variety of different problems.
And we find that very valuable for actually just seeing

how does someone attack a real problem
that they would be solving at work
and will they want to do that in a way that is curious,
digging into the details,
collaborative with their interviewer and so forth.
So that's the point.
Structured loops, they're consistent.
And then questions that are as close to the real job
as we possibly can.
And for the questions on the PM side,
do you have them do it at home
or do you have them do in the office?
It's a variety.
Right now, at Stripe, we're very much in a hybrid mode.
So a lot of our interviews are still happening over Zoom,
although we have plenty of great activity
going on in our offices as well.
The written exercises tend to be something
that you do in your own time.
We suggest a time bound,
so it's not an unreasonable demand
in the rest of your life.
And then the other exercises are within interviewer on Zoom.
Okay, so speaking of product managers at Stripe,
Stripe is kind of famous slash infamous
for waiting a long time to hire your first product manager.
From what I understand, it was like
around the 200th employee and maybe five years in,
which to me tells me that you're really good
at building product-minded engineers
or hiring and building a product-minded engineering culture.
How do you do that?
How do you go about doing that?
And then I'm also just curious
what PMs ended up bringing to the table
and how they kind of collaborate now.
Yeah, so I mean, as Stripe today,
we have lots of product managers, lots of engineers,
product management's an extremely valuable
and important job family.
I'll talk more about what folks do today.
It is true that our original team,
I think we hired our first PM a little bit earlier

than you said, but our original team
was really an engineering team.
However, they were extremely product-minded.
And I would honestly say that of all of the very early team
that I've worked closely with and known well over time,
every single one of them could also have been
and essentially also was acting as a PM.
If you think about the nature of the first Stripe product,
and again, today we solve problems for businesses
from the very small to the very large
across a whole range of financial infrastructure,
but the initial product was very developer-focused
and we still have a developer focus
at the core of many things that we do.
The best PMs for developer-focused products,
I think are usually very technical.
They're mostly former engineers themselves
or maybe current engineers still think bring on the side,
but bringing a lot of user insight
and strategy to the problem.
And every single early member of the team at Stripe
had to act like that in order to work the way that we do
and that we want to.
So remember again, that is to get to know our users very well
and then bring those kind of personal insights
into our product development loop.
So I mean, in fact, the way we think
about product development at Stripe,
it really is to find the correct set of early users
to kind of co-create the product with.
So maybe the best example of that
or at least a bad example of that is Stripe billing.
So it's our solution for subscriptions and invoicing.
And when we got to starting the Stripe billing product,
we realized that there were a number
of our existing users.
These were companies like Figma and Slack.
We were already using Stripe for payments
that had these subscriptions business models
and we figured that there were gonna be many more
of these kind of companies into the future.
And we could see that they were really kind of pushing
the boundaries of what was possible here.

So we decided to co-create the product with them.
The early team working on billing
got a bunch of those companies,
included Slack and Figma, but a bunch more as well,
got to know them that the individuals
who were operating those systems
that those companies personally understood
exactly what their challenges
and hopes and dreams for the future were
and then brought them into kind of shared product development
so we had shared Slack channels.
We'd actually show them product on a very regular basis,
get their feedback on it.
And only when that original kind of Alpha group
was super, super happy with the product
did we then think it might be ready
to go to a broader audience.
So that's just how we build product at Stripe.
And that means that every engineer
building product at Stripe really has many
of the kind of attributes and will exercise many
of the attributes that you often find in NPMs
in other companies.
That doesn't mean that there isn't a really important job
for PMs to do here as well.
So something else that I think is worth knowing
about Stripe and it's somewhat unique
is building product is also a much more
kind of team-sported Stripe across functions.
So there are the obvious functions
that most companies have.
Engineers, engineering managers,
product managers, designers,
and those folks all get involved in throughout
the entire lifecycle of product development at Stripe.
But also if you think about the nature of our products
it turns out that a lot of our products
are abstracting over what the financial system does at large.
So sometimes there's a capability in a certain country
and there's a different one in another country
and we want to make it all work consistently.
So our financial partnerships matter.
So sometimes folks from our partnerships team

are part of the early development phase of a product.
Similarly, when you think about product legal
and many of the other functions around risk
and compliance and so forth,
we actually need their creative thinking
in order to build the right stuff for our users.
So the point is it's a way more cross-functionally
collaborative process than I've seen
in a lot of other companies.
And PMs play just this absolute linchpin role
across a lot of that.
So PMs are very frequently the ones
that are providing the kind of locomotion
to bring all of that partnership together.
They're very frequently the folks that are synthesizing
what we're learning from our users.
Not everyone is talking to users
across all those job families,
but the PM is probably the person talking to folks
the very most and probably the one
who's synthesizing the very best.
They often and usually also bring
a lot of the product strategy.
So, okay, this is what we're doing today,
but what does it ladder up to?
And therefore, how do we make the right choices
to make sure that we're taking the right longer term path?
And so it's a super important and valuable role.
And I mean, you mentioned a couple of great ones
on the call here already,
but I think that the PMs at Stripe embodying a lot of
the culture I talked about in our operating principles,
they just contributed a tremendous amount
to what we get done.
Perfect segue to where I was gonna go
with the next question, which is,
one of your operating principles is
to be meticulous with your craft.
And I know this is something that you focus on a lot.
What I'm curious about is, that sounds great.
Everyone would love for that to be a focus of theirs
and a core value and the way they operate,
but there's always this trade off of like,

we just got to ship stuff,
like it doesn't need, how do we wait until it's perfect?
So what I wanna ask is just,
how do you operationalize this principle?
First of all, and then I'm gonna have
a few follow-up questions.
Yeah, great question.
Before I talk about this one operating principle,
let me just set a little bit of context on
what are our operating principles at all?
So at Stripe, we have operating principles
and they're kind of like corporate values,
but they're not just values.
They're much less abstract than that.
So actually we were just talking about the early Stripe team.
I think one of the best things that they did,
I mean, they did a bunch of great stuff
finding early product market fit,
but one of the most foresighted things that they did
was to kind of take a little detail and think about
what is it about how we've been working so far
that we think has made us successful
and is going to be important as we continue to scale.
And we capture those a set of operating principles.
So their behavior is distilled
from the most effective Stripes.
And we really care about them.
So we talk about our operating principles a lot internally.
I'll often see individual Stripes kind of celebrating
each other's work in terms of which operating principle
this kind of best represented.
We frame a lot of our feedback processes
and so forth around them, although not exclusively.
The very first operating principle we have, by the way,
is users first.
So we've already on this call touched a bunch
on how we put our users at the center
of understanding what we should be doing at all.
And that idea that we form these deep personal relationships
with our users to guide everything really comes from
and is captured by that operating principle.
We also have operating principles.
They're kind of segmented into how we work, who we are,

and then a bunch that the leaders must execute.
So I mean, another example of one of the how we work ones
is move with urgency and focus.
Like we really believe that even though we're building
infrastructure that will persist for decades,
it is important that we solve the needs
or users have right now, relatively rapidly
so that it actually makes sense
and is delivering value for them.
So anyway, there's a range of them.
You asked about being meticulous in our craft.
So this is another how we work principle
and it is very important as Stripe.
And you also call it, Lenny,
a kind of challenge of being meticulous, which is,
well, there's so much to get done.
So where can I be meticulous?
Because, well, if you do it across everything,
maybe a lot of progress would grant to a halt, right?
So, well, first of all, let me tell a couple of stories.
So before I joined Stripe, I spent some time
as a big part of deciding to join playing with the product
and something that really delighted me
was that when I started integrating the product,
there were parts of the experience where I was stuck, right?
So I made a mistake with how I'd integrate a BPI
and I was getting an error message back,
but I wasn't stuck for very long
because something that the team at Stripe had done
was we made the error messages coming out of our API
linked to the piece of the docs
that told you how to solve the problem, genius.
That is a really good example of being meticulous in the craft.
So it's pretty uncommon,
it's certainly very uncommon back then
for companies to do this,
like the error messages on your API,
who's ever gonna look at them?
Turns out developers, while they're building
and if they're running into problems,
they will look at them.
It's actually a really high stakes moment
because it's a time that matters.

So that is an example of where we'd be meticulous
in the product.
And the way that we kind of figure out
where to be meticulous,
like where to really sweat the details
and go above and beyond
is again, we try to be as users first as we possibly can.
So we have a process that is widely used
across product teams
that even like folks building stuff internally
for Stripes to use, which we call friction logging.
The way this works is you need to kind of put yourself
in a particular user's shoes.
Like it's actually important to have a very clear idea
of who is the person I am
kind of modeling the friction for right now.
So for instance, I might be integrating
the Stripe Billing API
and I might put myself in the shoes of an engineer
at, let's say, Atlassian,
which is none of the world's largest SaaS platforms
and using actively Stripe Billing
for automating all of the revenue today.
I might put myself in the mindset of that person.
I might have met them recently
and kind of understood them a little bit better
than otherwise, doing this particular thing.
And then I will go through the process
of actually using the product,
starting off in the dashboard,
hopping over to the particular page in the docs
that I need to follow,
starting to write code and so forth
in order to do my integration
and make really careful and meticulous notes
of what the experience is.
So stream of consciousness notes,
but paying particular attention
to the places where I ran into friction
that I think that that particular user
and mental modeling would find difficult.
And then those are the places
where we will tend to go and invest a tremendous amount

in really thinking how could we solve this problem
and then putting a lot of attention
into detail into making it right.
And by the way, the example I gave you
around the error messages,
there's actually more code in the jobs
that serve the Stripe API to handle those edge cases
than in the actual main flow.
And I think that's quite remarkable.
Most people wouldn't do that,
but it turns out not only was it something
I was impressed with,
but when I talk to Stripe users,
this is very frequently something they tell me,
delights them about the product.
It's a reason that they can adopt it more quickly.
And it's a reason that they keep adopting
other Stripe products
because they know that it's gonna be easier
than doing anything else.
So it really matters.
So the point is like,
we are intentional about where the places were
being particularly meticulous matters.
And then when we are,
we try to do it in quite a systematic way.
It's great business as well.
So for instance,
we actually just earlier today put a blog post up
that explains,
we've been tuning our,
both what we call our payment element,
which is our kind of morph kind of batteries
included UI for putting payment integrations
directly on your website.
You can style it to fit with the rest of your page,
but it has a lot of powerful features.
And we also have a set of hosted services
called Stripe Checkout.
And the point here is we've gone and scarred
a whole bunch of like checkout flows on the internet
over the course of many years.
And we look out for all of the little broken edges

and we've been working hard in these products
to tune those.
And then don't just stop there.
We've also been really obsessing about
what are the things that we can do
in the flows that we've already built
that will remove latency or remove a click or whatever.
And what we've found,
we've recently measured the difference between,
so with a bunch of actual users who've migrated
from a fairly vanilla integration
where they built their own checkout flow
to one of these surfaces to our payment element
or to Stripe Checkout,
it turns out that it increases the average user's revenue
by 10.5%.
And that's huge, right?
So this is an industry where little changes
that you might go about making
or even like big changes that you pour blood, sweat
and tears into will typically deliver uplift
in that you measure in basis points.
That's hundreds of a percentage point.
And this set of small changes company over time
ladders up to 10.5%.
It's huge.
And the point is we only get there
by knowing that this is a thing that might ultimately matter
and being meticulous in every single step along the way
and that it compounds to a dramatic impact in the end.
So I think that that particular set of experiences
is one where this operating principle has really mattered.
Thinking about those kind of flows
is what led us to build Link,
which is our product for making one click checkout available.
And that has had a tremendous impact,
both on convenience for customers of our users
and also on their conversion rate and so forth.
So it really matters.
And then one final other area, if I may,
I think Stripe is relatively well known
for putting a lot of care and attention into our website,
a lot of our marketing or product explanation pages.

And I think for a business like ours,
it's not immediately obvious
that it makes a tremendous amount of sense
to be super meticulous in building those experiences.
But we are, and again,
I think it really matters for us and for our users.
So the way that we got into this mode
was we recognized that we are building a lot of infrastructure.
So we built what we call our global payments and treasury network.
It is literally our product infrastructure
for moving money into Stripe,
holding balances on behalf of our users, moving money out,
all of the automation between moving money
between different businesses
and all of the kind of regulatory stuff we have to handle,
all handle in this big payments engine.
But as a user, it's very hard for you to kind of come in
and inspect that and say, was it good?
I mean, hopefully you can go and talk to a bunch of other Stripe users,
but it's very hard to know from the outside.
So if we put a lot of care for detail and attention
into the experiences that we use
to explain the value and the power of that,
it actually helps our users understand exactly
how it is that we are trying to operate for them.
And that's where all the meticulousness
that we put into our site have come from.
There are lots of features like we have a spinning globe
on our page that shows you what payment methods
you can use in different countries.
We have this kind of big animated wave on our homepage.
And in general, Stripe is kind of known for these great internet moments
when we launch products.
And so we did it because we wanted our users
to be able to see the care and attention that we were taking.
It's termed like to have this wonderful side benefit to some extent
because those experiences tend to get shared a lot,
like on Twitter and LinkedIn and elsewhere,
because they're worth looking at just because they're beautiful.
And they often push the web platform forward.
And that then means that folks who are likely,
folks who themselves want to push internet business forward
are likely to know about Stripe and know about our capabilities.

So meticulousness plays with impact in so many different ways at Stripe.
And I enjoy working that way in many different areas.
I have a few follow-up questions along those lines.
But before we do that, I want to double-click on this friction logging process.
Just understand how you actually do that.
So maybe two questions there.
Is this like a regular task people have on their plate?
Or is executives or just PMs in general?
And is there a template that you share just like you talked about?
Who are you? What company are you working for?
What problem are you trying to solve?
How do you actually kind of tactically, operationally, is that?
Yeah, great question.
So it's a practice that is quite valuable generically.
There are many places where you can apply friction logging
in order to really kind of shine a light
on what is the most effective place to invest time and effort.
So we use the practice across lots of different functions
and in lots of different ways at Stripe.
But it is the case that almost every product team has someone.
It's often the PM, someone who's just the engineering manager
who has a regular repeating loop of going through the end-to-end flow
of the product and writing a friction log.
For years and years, I have gone through the process
of onboarding as a new user to Stripe once a month writing
a friction log and then tagging in the right people across the company
that might need to take action on some of the things that we're observing.
And one of the reasons that that kind of process of taking a big step back
is useful is at this stage, we have many people.
We have thousands of engineers who are working in parallel.
And while everyone cares a lot about being meticulous,
paying attention to detail, getting the details right,
you can often end up with experiences that diverge
and going through this process of looking at it all together
on a regular rhythm with friction logging
really helps us maintain a cohesive whole while we all operate in parallel.
And so therefore, senior leaders, executives, I guess, of our larger areas
will often do this as a practice themselves
in order to kind of make sure that everything that they're responsible for
is coming together well.
So it happens kind of recursively at different layers.
So your question is, is there a template?
It's a relatively simple process.

So, yes, there is a template.
And in fact, there's a Striped Up Dev article
that maybe we can put into the show notes that has the template.
But it literally is, say what you want to do,
be very explicit about the user that you are trying to have a mental model for
because that helps you make the right choices as you go through the flow
and then really just keep a very clear
or stream of consciousness log of what you're finding as you go.
By the way, also really important, praise the stuff that's good.
So I'll often set these docs around to a lot of folks inside of the company
and it's a great opportunity to recognize great work done.
So if I ran into that error message that links this documentation,
it's a great example where I'll be like, this is awesome.
Nice work.
Being a PM that has received emails from the CEOs of companies
with all the problems they ran into when they're trying to book,
it's often like, oh, God, I have so many goals I got to hit
and I have this timeline, I have these things I'm working on.
What is the culture like, slash, how do you give team space to do these things
that are just like, we're just going to make the experience better?
How do you actually do that so that PMs aren't stressed out when they get these things?
It starts with our operating principles.
So because we have operating principles of users first
and being meticulous in our craft,
we actually tend, all of our ways of building plans tend to be wired quite well around
this idea that we're going to reserve enough time to really make the experience good.
Maybe the most pronounced version of that is not even the issues we identify in friction logs.
Maybe in a bit we can talk about how we invest in reliability and so forth,
but things go wrong and one of the things that's very important to me
is that we are a learning organization.
We need to understand why they went wrong and then take action
to stop the same thing going wrong in the future.
And so that means that we identify incident remediations.
That's what we call those and we prioritize those carefully
and most of them, the ones that matter,
actually get prioritized before other work on the roadmap,
which means that as a PM and we are building your roadmap and your plans,
you do have to think about, well, approximately how much bandwidth
do I need to reserve in this area in order to address
polish stuff that might come up through friction logging
and also to take care of those kind of operational things as well.
It varies by team and varies by stage of product.
So there is no stripe kind of like,

we will reserve 50% of our time to do this stuff.
That wouldn't make any sense,
but we do encourage and ask every team to think hard about
how much should they be setting aside for those kind of activities.
Does that make sense?
Absolutely. I was going to ask you if there's a percentage
and basically you trust the teams to set aside the right amount of time.
That's right.
Awesome.
Okay, so sort of along the same lines,
I actually asked Shreyas Doshi what I should ask you.
Oh, cool.
He had no idea I was going to do this.
And what he said is I should ask you how you did UX reviews,
pre-shipping, right after you joined Stripe,
and he just said you were very good at this.
And so what is that like and how should people learn from your experience?
The way that I did UX reviews very early on in my time at Stripe
and not the way that every group at Stripe does these reviews
is employing some of the techniques we talked about already.
So that active friction logging, it's very often done asynchronously.
So someone will sit down and reserve an afternoon
and go through the product and make particular notes.
And that's super useful.
We also find it very valuable to go through that process together.
So we will bring together both the team that built the product
that we're looking at, but also a lot of their cross-functional partners,
so things like the support folks that are going to be handling
or kind of building the process to handle questions from users about it,
executives in the org that they're a part of to do these walkthroughs
where we're taking exactly the same approaches I described with friction logging.
So imagining we're a particular user and experiencing the product together.
And then just discussing very openly,
the way that we do this today is we have a log of issues
that we want to talk about that anyone can type into
while the PM usually is walking us through the flow.
And then we'll get together at the end and actually talk about each issue
and does this need to be addressed? What do we really think about it?
So that's the approach that I took early on and that we take in general at scale today.
There's a lot of benefit to doing this together.
So someone once described this to me as if you were a chef,
you're going to taste the soup and you're going to talk to your kitchen staff
about what went into the soup when it was good and wasn't so good.

If you run a movie studio, you're going to sit down and watch a lot of movies together.
And so actually kind of looking at product together,
I find extremely energizing and also really valuable
in actually teaching some of the kind of bar for craft
and values around how we want the product to interoperate for our users
at scale across the company.
And building on that, there's something that we do occasionally
and I've done occasionally, it's called Walk the Store
where we'll actually do that process of like looking at the product together
with the whole company.
So we have a weekly meeting called Friday Fireside
where you don't have to attend, but majority of the company does attend.
And we've done a few series where we'll actually go through
some really interesting and critical product flows together
and talk about how this is reflecting various priorities that we have
and the shifts that we're trying to make,
but in particular with the user experience and a particular user at the core.
And that turns out to have a tremendous amount of benefit and value
in helping just everyone have a shared language for talking about things
and get on the same page about everything.
So yeah, those are some of the techniques that Trayas might have been thinking of.
It's amazing how many directions you all come at to create this high quality and product.
Walking the store, friction logging, these like entire company meetings,
looking at the product, UX reviews, like this is how it's possible
to build something so high quality, right?
It's not just way of this value and we're going to be building great stuff.
It's like you have to come at it from so many places.
Yeah, that's an interesting observation.
I think almost anything that you talk about is a value.
I mean, if being a value matters, but you need to have a practice behind that
that means that the value becomes real for everyone.
So we very frequently find that for any product development team,
so long as they are, they identify the right metrics
that actually represent the experience that they want to deliver for their users
and then get together frequently, like predictably,
to look at how the team is doing against those metrics.
But then everything else just happens naturally
because you understand what you're trying to move
and every little micro decision you're making within your own time prioritization
and then in terms of the actual work you're doing will ladder up to that.
And so when I say that, it is like you have to have the predictable
and regular thing that ladders up to the value you're trying to deliver
in order to make it happen.

Coming back to the UX review, just a question there,
presenting to the CTOs often very stressful in a review like that.
What advice would you give PMs and designers and just leads of a team
for how to prepare for a meeting like that, whether it's Stripe or anywhere?
I personally try to be as friendly and unscary as possible.
But I know that no matter how much I do that,
that these can be high stakes meetings.
But Stripe, I think it's probably for most companies, but Stripe,
the main answer here is put your user's hat on.
And if you understand your user well
and what they're seeking to get out of the experience and anchor any questions
you get asked or wobbles you may feel like that was an out of the blue question
back to, well, here's remember what we're trying to do for our users.
That usually makes any meeting like that go really well.
And so that would be my main piece of advice.
So I mean, a risk that exists in any company as you get to run a bigger team
or whatever is that individual contributors,
individuals will have a very small amount of time with you in general.
So there's always a risk that you might make a fairly unimportant remark
and it will be taken out of context to be something very important.
So I personally also try very hard to anchor feedback I'm giving
in what is the user experience we're trying to deliver?
Does this actually matter?
Recognizing that that is a risk and it definitely takes constant practice.
You've touched on a couple of these things.
But just if someone is trying to get better at building product,
either a PM designer engineer, what kind of advice do you often give
for helping people just build better product?
Almost always goes back to things we've already talked about here.
So remember at the very top, I talked about iterating very closely with users.
If you have a mechanism to listen to users,
get something in their hands quite quickly and then get their feedback on it
to run it back through a feedback loop, you're very unlikely to go wrong.
Especially if you put a lot of thought into these being the right users
to whose needs you want to focus on because they led up to your strategy.
It's actually very hard to go wrong if you have that feedback loop working really well.
So if you don't have that feedback loop, and actually it's remarkable
as I talk to folks across the industry and so forth,
it's remarkable how frequently that feedback loop doesn't exist
in product development cycles.
So if you don't have that, go figure out how to create it.
And if you do have it, but you can't get something into users' hands
very rapidly from you having the idea that this might matter

to getting their feedback on it, figure out how to make that go faster as well.
And at Stripe, we focus a lot on all of our developer tooling and infrastructure
on making it possible for changes to get delivered continuously to production
so that we can get them in front of users very rapidly.
And that's really important.
This episode is brought to you by BrainTrust,
where the world's most innovative companies go to find talent fast
so that they innovate faster.
Let's be honest, it's a lot of work to build a company.
And if you want to stay ahead of the game,
you need to be able to hire the right talent quickly and confidently.
BrainTrust is the first decentralized talent network
where you can find, hire and manage high quality contractors in engineering,
design and product for a fraction of the cost of agencies.
BrainTrust charges a flat rate of only 10 percent,
unlike agency fees of up to 70 percent,
so you can make your budget go four times further.
Plus, they're the only network that takes zero percent of what the talent makes.
So they're able to attract and retain the world's best tech talent.
Take it from DoorDash, Airbnb,
plaid and hundreds of other high growth startups
that have shaved their hiring process for months to weeks
at less than a quarter of the cost by hiring through BrainTrust Network
of 20,000 high quality, vetted candidates ready to work.
Whether you're looking to fill in gaps, upscale your staff
or build a team for that dream project that finally got funded,
contact BrainTrust and you'll get matched with three candidates in just 48 hours.
Visit usebraintrust.com slash Lenny or find them in my show notes for today's episode.
That's usebraintrust.com slash Lenny for when you need talent yesterday.
Something that I've heard about you is that you get into the weeds.
You get really deep as a CTO of a company at the scale of Stripe.
And I came across this term,
engineer occasions, which I think connects to this.
Can you talk about that and how you think about just getting in the weeds?
First of all, in terms of how much getting in the weeds matters at Stripe,
we find that it's very important for engineering managers in particular,
but really all managers to really have a very detailed understanding
of whether their teams are kind of on the right track
and where they're getting stuck in order to make sure that we make
the most progress in unit time for our users.
So we find that, again, the nature of the problems we're solving
really reward domain expertise, like assimilating a deep understanding.
And we ask all of our managers to be kind of the editors and chief of their teams.

And I think the only way to do that is to get the right loop for yourself
to understand what is actually going on in the ground.
Now, it would be really damaging if I was only doing work,
you know, the IC engineering level every day,
like that doesn't ladder up to someone who's able to like help guide the team or whatever.
So it's important to do this judiciously, but done on the right cadence.
I think it's very, very valuable.
So engineering occasion is something that I personally do.
So it's obviously a portmanteau of engineer and vacation, but it's not a vacation.
The way we get to the name of the chair, or at least the way I think about it
is what you do on engineering occasion is I'll clear other people
structure this as well, clear several days in a row, three or four days,
actually join a team, pick up a small task, hopefully a small feature
that we can get all the way from start to finish, like in production
and do that going through the exact experience the team has.
So you get to understand the developer tools, the build infrastructure,
how you get stuff reviewed, how good is the documentation?
How long did I have to wait in order to actually, you know, see my thing live
in order to start that feedback loop that I talked about with users?
So you're actually going and joining a team and doing some work while one does that.
It's really valuable and important to keep a friction log,
because then what you want to do is write up the experience
both as an aid memoir for yourself, because it then helps you go and, you know,
for all of the next years worth of conversations that might be involved
in Iran's making trade-offs and helping set priorities, that context really helps.
So I actually read these reports periodically.
It's also really valuable to actually share the work with the team, which then demonstrates
why I understand what it's actually like.
And here's how we're going to kind of make sure that we're kind
of getting that information through into prioritization.
So that's what you do.
The name is it's often the hardest thing about these is finding the time
to clear your calendar for that many days.
So the reason I think about it is like a vacation is of course people go on vacation.
By the way, I work very hard to use all my vacation days every year.
I think that taking a break and recharging your creative juices is valuable.
So the point is like when you're on vacation, the world goes on and it's all fine.
So I treat it like I'll decline every meeting in order to clear
my schedule and have a maker schedule in order to do that.
And so I've done this many times at Stripe.
I advise engineering managers at Stripe that they should do an
education sometime in their first quarter to six months at the company.
It gives you a tremendous amount of context for what your teams are actually experiencing.

And then for folks who do it once a year on an ongoing basis,
and it provides a tremendous amount of context.
That is insane.
I've never heard of that process.
How do you stay knowledgeable and up to date on the infrastructure
and the code that you have to code in all of a sudden?
It makes me so fast.
The process is literally to help you do that.
I guess maybe something I left out and explaining it is
we'll often identify a buddy on the team who's going to like help you through.
So for what it's worth at Stripe, we write a lot of code in Ruby.
We write a lot of code in other languages like Java and TypeScript as well.
Our core infrastructure is mostly implemented in Ruby.
When I started, I had never written a line of Ruby in my life.
And I knew I wanted to do this thing and actually done something like it before joining.
And I was really scared.
Like what kind of like how much credibility might I lose if I show up
and I can't write a line of Ruby.
But my first buddy was a guy called Akshay who's still at Stripe doing great stuff.
And he helped me learn Ruby during these three days.
Yeah, they rid me a little bit.
But ultimately, everyone really appreciated that I've done it.
Imagine being the engineer as to review your PR.
I tell them this is an important part of the process.
Please don't treat me any different to anyone else.
And, you know, obviously, to make this work well, you have to set the
expectation that you're not going to like take me action.
If something isn't exactly what you want it to be, it's a more
kind of longitudinal process.
Is there something that you found in this process that comes to mind that was
really surprising or interesting or just kind of lasting?
In the past, I don't know, a couple of years.
One of the most interesting things that I think I've run into is there are a
lot of places that as we scale, we know that it makes a lot of sense to invest
in automation and there were a bunch of places in our development process where
while we've done that, we were working hard to help each other out and actually
saying, hey, please come talk to this other team if you want to use this thing.
And if you're working, for instance, in a different time zone from someone else,
it's often the case that actually it would be much better just to document it
well and automate it and then maybe make the consultation available async.
And so like running into those kind of pieces of friction and then, you know,
having a good conversation about it has really helped.
But one good thing about Stripe, I think, is that kind of change doesn't depend

on me doing anything.
So we really care about making Stripe a place where engineers can do some of
the best work of their careers.
And that means being enabled with good tools and good developer productivity.
So we invest a lot in developer productivity.
We have a team who work on our DevTools and they actually run exactly the same
product development process that I just described for our external users internally.
So understand your users really well, talk to them a lot.
And so, for instance, we do a monthly survey.
We operate in enough skill now where we can ping enough people once a month
that we get a statistically significant sample of the entire organization
without having to get everyone to respond, every individual
new response by once every six months.
And we ask, you know, very directly, what's the experience you're having?
And we use that to prioritize the roadmap of our developer productivity team.
We also measure everything.
So we know where people are getting stuck.
And when we compare both kind of free responses and the data, it helps
us invest in the best places to make everyone else more productive.
Okay, that's exactly where I was hoping to go next.
And let me set this question up.
So I saw a tweet of yours recently where I have some notes here.
I'm going to read that you deploy changes to your core API 16.4 times a day on average.
Your uptime has been 99.999%.
And one in 10 people in the world have transacted with a business
powered by Stripe at this point.
And so my question is just, what does it take to achieve something like this,
especially in terms of tooling and culture and everything else you're just talking about?
Well, I think it's worth saying at Stripe, we are trying to do something I think
relatively unique, which is what we power for the businesses that build on Stripe,
our users, it's really business critical to them.
Like we're literally talking about money coming into your business to help you run
your operations or make it even possible for you to run your operations and maybe
pay your employees and so forth.
All the revenue automation that we do around billing subscriptions, our financial
automation products, helping you close your books like this stuff matters.
You cannot get it off.
And we also operate Stripe today.
Like we move as much money as all of e-commerce was when Stripe got started.
So we operate at a very significant scale.
And so business critical, significant scale.
We just have a tremendous duty to our users to get this right and be extremely
reliable and available for them.

So we do think about that a lot.
Now, there's one way to be very reliable, which is to try to change
things as infrequently as possible, like never change anything.
Then you have many fewer opportunities for things to break.
But we don't take that approach.
The needs of our users are evolving so rapidly.
The number of ways that we want to serve them and can serve them is evolving
rapidly enough that it matters a lot that we can operate in a kind of
constantly changing environment.
Hopefully I've explained like why that matters.
Like a type feedback that users only possible if you can actually get
something in their hands.
So we choose to design the way we work to hold those two things
true at the same time.
And it does take a lot of care and attention.
And it takes a lot of systems.
So maybe just kind of build this up.
First of all, there's a lot in our development process and the ways that
we take changes into our products that makes this possible.
One of those is we really care a lot about having really good test suites.
So we believe in automated testing.
We don't have manual testers because manual testers couldn't possibly
cover the vast array of API endpoints and configurations that we offer.
But automated test can.
So we work hard to have a lot of automated test coverage.
And then every single change that an engineer produces gets run through
this battery of tests before it can even possibly go towards production.
And then we work very hard as changes end up in production to put them
through increasingly realistic and then more broadly exposed environments.
So we have a bunch of staging environments where we'll like send
a battery of like more realistic end to end tests.
And then finally, when something actually goes to production, it starts
at very small percentage of the traffic and then wraps up to the whole.
So we can detect problems before they become a huge problems.
There's a number of things that we had to do to make that all possible.
So for instance, every change that a Stripe engineer submits
once it's passed those tests, it actually ends up in production automatically
over the course of the next 45 minutes or so.
And that I don't think there are a lot of financial services companies,
at least not until maybe the last few years that have operated that way.
And so that takes a certain mind shift.
You actually have to assume that that's going to happen and put
the right systems and processes in place.

And then the other thing that's important is to recognize that we just
we have to obsess about getting very systematically good at addressing
anything that can go wrong.
So I mentioned earlier, it's really important to me that we are a
continuously learning organization.
And something else that matters, of course, is things will go wrong.
You know, sometimes there is a downstream partner where something breaks.
Other times there is, you know, a particular kind of network
out of sort of our control or whatever.
So things will go wrong.
So it's important that we have the right systems in place that minimize
the damage that any individual thing going wrong can cause.
And we work hard on that.
We have redundant systems in a bunch of places.
We think hard about how something that breaks for one user wouldn't
kind of carry over into other users.
And then we actually very actively work hard to put things right when they are wrong.
So that's called instant response at most companies, including Stripe.
I actually think Stripe is very, very good at instant response.
We've got very good tools for both declaring incidents
and then pulling the right people together to put them right.
But we don't stop there.
We really obsess about reviewing them carefully and identifying not only
like what would have stopped this thing happening, but how could we
prevent this whole class of issues in future?
And as I said earlier, we prioritize working on that stuff
ahead of anything else on the roadmap.
That's because of remember what it is we're trying to do, how important it is
and how if we don't do that, well, we can't move quickly for our users.
So that's how we do it.
But I don't want to, I want to kind of come across here and sound like
we've got it all figured out.
Of course, all of this is always entirely, you know, in flux.
We're always anxious to figure out how we can make it go better.
For instance, in recent years, we realized that we could get a lot
of benefit from what we call chaos testing.
That's like injecting errors and making sure that the systems respond
in such a way that it doesn't cause any impact on users.
So it's constantly evolving and we, you know, we really enjoy learning
from other companies and learning from our users as well.
But it's something we care a tremendous amount about
and think very rigorously and systematically about.
Did you say that it takes only 45 minutes from pushing code to it

going into production?
Typically, yes.
So we, we, that battery of tests that I described that gets run on every change.
That gets run in parallel to when you send it for review by another human.
So it's run once, typically takes about 15 minutes.
And then once the changes merged into our code base, we run that same test suite one more time.
So another 15 minutes and typically takes about 30 minutes for the systems to automatically deploy into production.
That's, that's how we run.
Um, and, you know, think about establishing that tight feedback loop with users.
That means you can get feedback from a user in the morning.
You can figure out how to address it and you can actually put something back in their hands by the end of the day.
And that loop running inside of 24 hours, I think it's pretty important.
That's incredible.
I remember times that are being beware took hours for tests to finish because they just, there's a lot of them and they fix that over time, but I'm used to more on that scale.
It takes constant effort to hold those, those numbers, by the way, are important.
Like those are about the right numbers, I think, for a company operating our kind of scale.
Of course, as you add more stuff, you add more tests.
So we've had to work on, we've had to invent mechanisms to make it run more in parallel.
We now have a thing called selective test execution where we figure out for they, not the battery that runs before it goes into production, that we run everything and just run it in more machines to make it parallelize.
But for the changes, the, the tests that we run for individual changes, we actually have systems that figure out, well, which tests are the ones that might be material for this particular thing.
And that's been like a real source of innovation.
In fact, our, our distributed change and test environment that Stripe is the biggest distributed system that we actually run.
And so running that is, you know, requires just as much energy and, and effort and rigor and, and craft as everything else.
What are some of the things you've done that have had the most impact on developer productivity?
We've touched on a couple of these already.
That, that auto deploy mechanism definitely had a very significant individual impact.
So before we introduced that, so that sometime within the last five years, each deployed to production required an engineer to actually babysit it, to like, watch it as it went out, make sure all the charts look good.

And it was a bit of, it was a bit of a kind of game of roulette because you
always wish that your change would get bundled with someone else's so that
they were paying attention to it for you.
So literally being able to be like fingers on keyboard at something else while
your change is going on and it's going to automatically monitor that.
That was a big improvement in our productivity.
Also, like very small things that this is just like optimizing check
out very small changes really compounded in a big way.
So for instance, about eight months ago, we made it possible.
We've got to use a tool for code review.
It's just quite popular as GitHub enterprise, but we built a bunch of
our own experiences and controls around that.
We will tick box that you can take on your, on your pool request.
That's what you call an individual change that an engineer produces.
That is auto merge, which means once the reviewer says this is good, I
don't need to come back and look at it again.
The system should just take over.
And that just cuts out one whole kind of human distraction step.
And that laddered up to a big change in productivity as well.
So these very small changes can, if you're deliberate with them,
let her up to a big impact.
I'm going to go in a different direction now.
AI, very hot right now.
Yeah.
Has AI impacted the way you all build product yet?
And if not, where do you think it starts to go in the next few years?
Yeah, we're very excited about AI.
Now, to just take one step back, Stripe has been using machine learning and,
you know, advanced machine learning techniques at the heart of our
product since the very early days.
Radar is our solutions for payment fraud.
And it has used, I mean, this is the very core of our products and has
used machine learning since we introduced it.
We also, do you think about the nature of Stripe?
We operate an environment where we have to do a lot of work to kind of
catch bad actors in the system, fraudsters or fraudulent businesses.
So we've been applying a lot of machine learning techniques there again for years
and, you know, we couldn't operate the company without them.
So, you know, the difference between ML and AI, I think when we think about AI today,
we're mostly talking about the applications of large language models.
But those are really based on this new technology called Transformers.
It was a paper that came out of Google a while back, about four years ago, five years ago.
And we put Transformer technology into those systems at some point more than a year ago.

And it's proved to have a very profound impact on our ability to solve those problems for our users, which is great.
But we are also excited about the applications of large language models.
I mean, there's really two dimensions to this.
Number one, we feel privileged to be able to help serve and power a lot of businesses getting started in this space.
One big difference between AI startups and others is it's actually quite expensive to operate an AI startup in terms of the compute resources.
Like running this stuff in these GPU machines is quite expensive.
So typically, these companies need to have a monetization model from day one.
And I'm really excited to say like most of the AI economy is running on Stripe.
And we're working very hard to make sure we serve all their needs.
Open AI, for instance, is using us for monetizing chat GPT plus and all their other products, but they don't just stop there.
We're actually helping power all of their subscriptions and revenue tracking and financial operations around the business.
And the point about that is we've been putting very large engineering teams on this stuff for years, and that means, you know, anyone who wants to build a reconciliation product and a subscriptions business model, if you want to do it yourself, you have to put, you know, big engineering teams on them.
But if you can use Stripe instead, you can actually focus those really precious engineering resources on keeping up with the rate of, you know, breakneck innovation in this space.
So we're excited about that.
But we're also really excited about the applications of these AI techniques on our own ability to serve our users better.
And so, for instance, at sessions, we'll be talking about a few of these.
We we started working with Open AI when they had the data.
It was a kind of private beta of GPT for available at the beginning of this year.
The first thing we thought of was we've a lot of documentation on Stripe.
And we put a lot of care and attention into making it good.
But if you want to achieve something with your Stripe integration, you're typically going to spend quite a lot of time reading docs and kind of synthesizing them as an end user.
We realized we can have GPT for read all of our docs, store them as embeddings and then answer questions for developers.
And so we now have that available in early stage release inside of our documentation.
It's turning out to be pretty valuable.
We've also been able to apply these techniques to other areas of our products.
So something that we're going to show an early version of at sessions is we have a really powerful product as part of our revenue and financial automation suite, which we call Sigma.
It allows you to write SQL queries across all of your Stripe data.

So you can like interrogate your Stripe data to understand your business
in really fine grain detail, you know, which country has the fastest going
sales, which segment in which country has, you know, particular attributes to it.
Very powerful product, but it's one that is potentially challenging for
non-developers to adopt.
Like you do have to know how to write SQL queries, unless you just use the
ones in the menu in order to get the full value out of the product.
Well, it turns out with large language models, we can apply this technology
where you can ask questions in natural language and our engine will actually
write the SQL query for you.
And we've had to do a lot of work to kind of tune that to make sure that
it's reliable and understandable and it's going really well.
And we also see great applications in actually applying these technologies to
make us more efficient internally, answer users questions faster, help each
other out more quickly.
And so we're doing those things as well.
One concrete thing that we did, which I'm pretty excited about is not long
after we saw a chat GPT come out, we realized it would be really awesome
if we could apply that to many use cases inside of Stripe.
But as you can imagine, a lot of the data that we are dealing with on a
daily basis is very sensitive to our business or users.
And we care a lot and we have a lot of governance around this, but we wanted
to make that technology available.
We couldn't say, hey, Stripe, just go and use chat GPT.
So we stood up and integration to GPT for and an internal UI to use models
like that.
But here's the thing that I want to kind of relate to all of your viewers.
We find that we built presets into that.
So when you work with large language models, you want to write a prompt
that then helps get the model into a state where it's going to be able to
solve the particular problem at hand.
And we find that writing prompts is something that's accessible to folks
across lots of different job families, not just engineers, folks in our
marketing team, folks in our user support team have been able to use this as well.
But you can put a lot of care and attention to getting a prompt that works
really well, and we now make it possible for those to be shared across the company.
So I can grab the preset to help me write, you know, a blog post with the right
kind of tone, and I find it extremely valuable in helping me and many
other people across the company.
So I think there's a lot, a lot of innovation in high companies serve
their users and operate as possible right now.
And we are working hard to employ all of that on behalf of our users.
What about as engineers?

Do you are you pro co-pilot?
Is there some along those lines of you?
Find useful, where you have, how do you think about that?
We're excited about co-pilot.
We, we've made co-pilot available to our engineers internally.
We ran a fairly rigorous trial to make sure that we felt good about, about its actual, you know, impact on our ability to write code and the code it was actually generating and find that it was very effective for us.
Both in productivity, but also just in helping our engineers feel, feel like they could direct their energy towards some of the bigger problems like how should the stuff fit together rather than some of the micro problems.
And so because we had such great feedback on that, we've, we've made it available, uh, very broadly internally.
And, uh, we'll be very excited to see where that, that space develops over time.
Do you have a sense of the productivity gain that say, I imagine you have a lot of 10x engineers.
The gains that an amazing engineer gets out of a co-pilot versus, you know, newer engineer.
It's honestly too early to tell because we've only recently made it available at scale.
What we're typically seeing is that the kind of tasks that it accelerates are, um, remember I mentioned how much energy we put into having comprehensive test suites.
It's actually using co-pilot to generate your test cases.
Turns out to be extremely valuable.
Uh, there's, there's a reasonable amount of boilerplate that is kind of like repeated code and concepts that goes into writing good tests.
But you also have to reason very carefully about, is this actually testing as a co-pilot one?
So co-pilot takes a lot of that boilerplate generation out of the way.
So you're actually thinking about the stuff that really matters.
And I think that is going to have a profound impact on quality of code that we write as well.
But it's a little too early to say, because we literally just will divide what kind of actual measurable impact is having.
I was talking to an engineer and he's just like, I like the act of writing code and it makes me sad that this is doing this for me.
So he, he turns it off.
But I think over time people will be like, eh, that's straight.
I never really enjoyed this part of it.
I mean, I'll speak from personal experience.
I love writing code.

Like I really love writing code.
It's pretty much what I do for fun.
I also love writing code with co-pilot because as I said, it means that I can focus on the parts that really matter.
And I think that's been a fairly consistent experience across Stripe as well.
Awesome.
So you run a massive engineering team.
I don't know if you share the numbers of engineers, but I know there are many.
What are, and this is a broad question, just what are some lessons you've learned about managing people and or managing engineers, whichever direction you want to go?
Wow.
Big question.
There, there are so many different directions we could take that.
So I'll, I'll share a few observations.
They don't necessarily all have a particular theme to them.
I think as I've been responsible for bigger and bigger teams over time, um, one of the things that, you know, just repeatedly learn is I, I personally will not be involved in really any of the decisions that matter and happen.
Like there are, there are thousands of decisions that an organization of any skill is making every single day.
Every individual engineer or PM is making hundreds of small decisions and some big ones every day that affect the kind of general trajectory.
So the most important thing is to really focus on hiring the right people.
And that means like hiring people that you can trust with a tremendous amount of autonomy, like if I try to get involved in lots of decisions, everything will grind to a halt.
So how do you do that?
It is important to be rigorous.
We've described already what our hiring process is.
I really lean into getting to know folks in, uh, you know, very, very well as, as we're hiring them.
By the way, something I didn't mention earlier that is, I think important here is we pay a lot of attention at strike during the hiring process to references.
This is typically later in the process, but, you know, if you put someone through an interview process, you've probably spent what, like eight hours total with them, if you talk to people that actually have worked with them before, you're probably benefiting from thousands of hours of direct experience.
So like we take this very seriously and we, we try to ask, you know, smart questions that will help get real signal out of that.
I've certainly found when it comes to hiring folks that have then gone on to really, you know, make a transformative impact that references are often the time that you get the best conviction on the hire or not.

So hire the right people and then you have to trust them.
My trust is interesting.
It was actually quite challenging to trust people that you've just hired, right?
Well, you've got no stake really on you yet.
Um, so what I find is you do actually just have to be generous in giving trust
and assuming trust by default at the beginning, but then you do need to,
you know, hold people accountable enough that they are proving that they,
uh, you know, they, they can handle that trust.
So, you know, sometimes that means hiring someone who you think quite plausibly
can end up in a very big role into a smaller role to begin with.
Other times you just have the big role you need to fill.
In which case, um, it's important to like be quite conscious of it, really
trusting and delegating to them with, you know, a lot of support.
So I'm always working hard to delegate sometimes like a little bit more
than I'm comfortable with, because that's the only way to like really
operate at significant scale.
I think something else, uh, I'm sorry, this is really a smorgasbord.
It's perfect.
And something else I've realized as the organizations I work with
and being responsible for have grown is that you have to be extremely conscious
about managing your own time.
There are hundreds or thousands of people who like quite legitimately might want
to get some time with you.
And, you know, if you just let your places that you spend your time and energy
be controlled by the stuff that comes into your inbox or the stuff that gets
put on your calendar, that's going to be random.
So, um, it's quite likely that that's not laddering up to the biggest impact.
So I personally have been running a look for, I don't know, maybe like 10 years
where on Sunday evenings, usually, well, it usually starts in Sunday
afternoons, I'll kind of read a lot of what happened last week.
And then a Sunday evening is actually write a list for myself of, um, you ask
me a question that started this podcast, which was like, what would, um, what
would like be great success for this podcast from your perspective?
I basically think about that for my week, every Sunday night.
I make a list of if we got these things done this week, that is a good week.
And obviously we need to be dynamic.
Like things will surprise you through the week.
But honestly, then that, that drives a lot of just where I decide to spend my time.
And, you know, I try to, you know, encourage everyone to think that way
throughout the organization.
And I think it does let her up to more impact over time.
I think the final thing I'll say here is I mentioned how important operating
principles are to the culture.

I think for any manager and any leaders, that includes ICPMs as well, but for any manager, any leader, like how you show up really does set the culture that, that is around you.
So I try very hard to show up very consistently every day, even though there will very frequently be things that are going wrong and are bad and are hard.
It's actually very important to show up quite consistently, certainly, you know, in line with all the values that we have and the culture that we want to set in order to really kind of model that at scale.
And, you know, different days, that can be easier and harder.
So the final thing is, I think it's quite important to manage your own energy.
There are some tasks that I do that are maybe not the most important thing, but I know that I get a lot of joy and energy from them.
And then that carries over into the other stuff that also needs to get done.
Just two more questions before we get to our very exciting lightning round.
One is just, I feel like Stripe is incredibly good at planning and organizing, prioritizing. I know every inside it's always not as beautiful as it may seem on the outside, but what have you learned?
What has Stripe learned about planning and prioritizing that you think other companies might be missing?
Yeah, well, it's funny you say that.
I would say planning inside of Stripe does not get the highest net promoter score that you might imagine.
I do think it is actually quite effective.
And by the way, the reason I think that planning, the way we do planning at Stripe doesn't always get the best internal wrap is we have grown very rapidly.
And that means that almost every time we come back around to planning, we actually, and this is kind of a theme, we think about a lot of the ways that we work internally from first principles.
So we will tend not to, by default, just pick a system off of the shelf that some other company has run or that we've used before.
We'll think hard what considering what we're doing for our users, how should we do this?
And when we're thinking of first principles, we'll often go out and talk to a lot of other companies and try to learn from their experiences.
Actually, something I've really appreciated and enjoyed at Stripe.
A lot of the learning I've done here has come not only from folks that are at the company, but also talking to a bunch of folks that we have the privilege to work closely with because their business is building on Stripe or, you know, they're at a similar stage or scale to us.
I mean, a good example of this is Amazon.
Amazon is a Stripe user because we work with them.
In my first couple of weeks at Stripe, I was actually able to meet with Charlie Bell, who then ran operations at Stripe.

And a lot of the things that, sorry, who then ran operations at Amazon,
a lot of the things that we actually now do at Stripe have come from learning
from that experience, but it wasn't just, oh, yeah, we'll take the thing
because Amazon did it, talk to him and lots of other companies in order to learn.
So the point is like we care about learning from others and then we apply it.
So bringing this back to planning, because we tend to think a lot from
first principles and we've been growing rapidly, the right planning process
for us as a company has actually changed quite dramatically
and quite significantly over the years.
And so, you know, we do planning in a relatively deep way once a year
and then in a slightly lighter way halfway through the year.
So if you imagine you do something once a year and you're growing rapidly,
you're going to want to revisit exactly how you do it.
And so I think, you know, for anyone who has the privilege of running
the same planning process again and again, you can really like
iterate it and tune it.
We've had to make more deliberate and kind of more sweeping changes
as we've gone along.
But there are a couple of things at the core of how we do this.
So one is it's really important to focus on who are the users
that we are seeking to serve for any given product area that we're working in.
And kind of holding their needs front and center in your plans is important.
Again, it's trying to do something that is it's not unconventional,
but it's like not it's not the most common practice in the industry,
which is we work very hard to serve companies from the very, very small,
like the folks that are literally just starting their company on Stripe
with Stripe Atlas all the way up to multinational Fortune 10,
Fortune 5 companies, Fortune 1 companies, I guess.
There's a very different set of user needs across those businesses.
So for any given area, it's very important for each team to be clear
about like who are they serving?
And then we run a kind of inverted W process.
So we typically have teams surface what their immediate thoughts are
on the most important things to do.
We'll then have a group of product leaders get together
and try to synthesize the most important parts of that into a kind of draft
overall company strategy and take that back down to teams to figure out,
well, how does if that's where we're making a big push, should that tweak
my plans at all?
And we bring it back up for synthesis and then back down for everyone
to really distribute with a lot of context within their orgs.
And we find that in our current scale, very, very effective.
I haven't mentioned this yet, but I actually use Stripe for my newsletter.

I check the app many times a day.
So I'm probably in the Fortune one million somewhere in that.
Do you have any feedback?
Genuinely, I'd love to hear your Stripe feedback.
It's, you know, that's great.
Like, the app tells me that I'm...
No, no, but don't tell us this great.
Like, what bugs you?
Well, it's interesting for churn, for cohort retention,
which I care a lot about for the newsletter.
I actually use a different tool that feels like gives me better,
a table of per cohort, how many people are still around.
So there's an opportunity to have better cohort retention metrics.
Cool. Okay, maybe we can email about that.
Quite genuinely, at Stripe, anytime we talk to a Stripe user,
we are always looking for feedback.
That goes for any occasion.
We often bring users into the beginning of our Friday Fire site
I already mentioned and we're always asking them for their feedback
and we really, really want to act on it.
Well, I wish I had more.
It works great.
Final question, Stripe sessions coming up.
It's going to happen before this comes out.
What should folks be watching for?
What's next for Stripe?
So we touched on a few of the things that we're sharing broadly
for the first time at Stripe sessions on this call already.
Per our development process, like, these have been in the hands of
some Stripe users for, you know, a significant amount of time
and we've really kind of crafted them with them.
But I'm super excited about a number of the things
that we're going to be talking about next week.
One is, I've mentioned our revenue and financial automation suite.
We've really been like crafting the features there for some time
and really nice example actually of how building with closely with users
really, really helps kind of build things at scale.
We've just set of features that we work with companies like Atlassian
and Cloudflare to enable their businesses on Stripe billing.
So they have relatively complex models.
For instance, you might sign a deal where it has a discount
in the first year and then there's another product
that's being used in the second year

and then something else happens in the third year.
You can now model all of that in Stripe billing
and we're going to be showing how you can do that
in the dashboard at Stripe sessions next week.
So it's now going to general availability for all Stripe users.
So it's giving, you know, all of our users from the very small
to the very large the same power that any, you know,
the world's best SaaS platform might have.
Excited about that.
It's also the case that we didn't talk about in this call.
Stripe Connect is our product for platforms and marketplaces.
And if I may actually just take one second,
it's a really good example of how focusing on the needs
of our users in one area kind of took us
into a really valuable space to solve for many users
in other areas.
So the ideas behind Stripe Connect came from working
with companies like Lyft and Shopify in the very early days.
They were using Stripe for pay ins,
but we realized all these companies are operating
in these multi-sided marketplaces
and there's a ton of heavy lifting you have to do
to make that work really well.
You have to have a bunch of regulatory licenses in place,
but also there's a lot that you need to do
to actually make the money move
and kind of account for it in the right way.
So Connect is our product for platforms and marketplaces.
And one of the things that we've done over the course
of the last year or so is actually taken a lot of the features
that we built in the Stripe dashboard
to help folks manage things like gathering the right documents
from their users or handling refunds and so forth.
And we've actually made them available
as embeddable UI components
that are also completely customizable,
which these platforms can take and put
inside their own dashboards.
So they can present a lot of the power
that Stripe is enabling to their customers
without having to do a ton of engineering work themselves.
Generating engineering leverage for our customers
is just very frequently the main thing

that we can do to help their businesses.
And we're going to show a bunch of that stuff next week,
which I'm excited about.
And then finally, we touched on it already,
but it'll be great to show some of the innovation
we've had around AI and large language models,
because again, I think it's going to make a big impact
over the coming years.
Well, with that, we've reached our very exciting lightning round.
First question, what are two or three books
that you recommended most to other people?
So most means that you have to integrate under the curve.
So I'd say High Output Management by Andy Grove
is definitely the book that I've recommended the most.
It was a book that really opened my eyes
to how to get the most out of teams over the whole course
of my career.
So definitely check that out if you haven't read it already.
Recently, so we talked about being meticulous in our craft.
I've really enjoyed Build by Tony Fidel.
Tony worked on the iPod and then the iPhone
and then was the founder of Nest.
And he puts a tremendous amount of thought into his users
on how to build really great crafted experiences.
I had the privilege of working with him very briefly at Google,
but I felt the book was excellent in terms of helping
provide some practical pointers there.
And then I would be remiss if I did not mention
Claire Hughes Johnson's book, Skilling People.
Even though it only just came out, I'd recommend it a lot.
There's actually some stuff in here
where I'm sharing some techniques that we've used with Stripe.
So you might enjoy that.
And then if I may share one, you've got it too.
It supports my laptop on here.
I hope you read it.
I've had her on the podcast.
I've read, you know, I've skimmed it as much as I can
because I'm not building a business right now.
And she was in the top 10 bestseller list
on the Wall Street Journal I saw.
That's right.
That's right.

It has a ton of very practical advice.
So I recommend that one.
What's a favorite recent movie or TV show?
Okay.
I want to cheat here because I'm going to interpret that as
like what's the best video content you've seen recently.
I have fallen in love with YouTube for learning about
like the really fast moving AI space.
So it has been remarkably valuable.
And you know, Andre Karpathy has a bunch of really good
seminars, but not just that across the whole spectrum.
I think there's just so much gold on YouTube
if you want to learn a new skill these days.
Is that the channel you recommend, Andre Karpathy's channel?
His channel is awesome if you want to learn about AI.
And he's the ex-Tesla AI.
That's right.
I believe recently I've gone to open AI,
but was doing this kind of on his own for a few months
and has produced a bunch of great YouTube content.
Great pick.
Most people pick white lotus or something like that.
And you go with the AI YouTube channel.
I love it.
What's a favorite interview question you like to ask?
Okay.
Well, actually you will find some recommended interview
questions for me in this book.
What I really like actually, because it sounds like a softball
and then it's not, is I often ask people, especially
for leadership hiring, which leader that they've worked with,
they admire most and why.
So it sounds like a softball, right?
But it actually tells you a lot about what this person
like really cares about in leadership.
Sometimes I'll follow up and ask like,
how does that manifest in your own leadership style?
But then, and this I think is really quite telling,
I always ask folks, I'm going to spoil my interview question
I've revealed in the podcast, but I always ask folks,
okay, so imagine you were their manager,
tell me the performance review or the development feedback
you give them to help them be more effective.

And I think everyone always has things that they could improve.
Certainly myself included.
And so folks' ability to think critically about someone
that they admire or lie in eyes
and how they can actually be more effective,
I find quite telling.
What are some favorite products you recently discovered
that you love?
The products that I have recently discovered
and definitely love is Mid Journey,
also building the business side of their business on Stripe.
But so for folks who don't know,
Mid Journey is an AI tool for generating images
using stable diffusion.
But it's really pretty awesome.
I've been using it a lot with my daughter.
So we'll come up with stories
and we'll generate beautiful looking images with Mid Journey
and then she'll drop them into books
and she'll write the pros.
The reason I think it's pretty cool is
I was very surprised and skeptical at its UI to begin with
because its UI is Discord.
They drop you into Discord and you're in a channel
where you have to prompt the artificial intelligence.
And I find it very confusing at the beginning.
It's like, this is not the right interface for this tool,
but actually it's very smart
because you learn from other people on Discord
how they're prompting the AI
in order to get the kind of results
that they're looking for.
And I find that made it possible for me
to get a lot more power and value out of the tool.
So I definitely subscribed to Mid Journey
and have had a lot of fun playing with it.
What's your favorite image that you've created
if there's one that comes to mind?
We made a cover for a book that my daughter is writing.
My daughter is nine years old, by the way.
So we're talking here, good children's stories.
She made an image of a wolf
wearing a purple velvet cloak

sitting in front of a campfire with a shack in a forest
and a nebula behind.
It's beautiful.
And we actually built that by combining two images,
the wolf and the shack.
And actually getting the wolf out of one image
and putting it onto the other,
we used the new Apple image segmentation company
and paste thing, which worked great.
So it's also fun combining these AI tools
but that one has been pretty good.
What a world.
Two final questions.
What's something relatively minor
you've changed in your product development process
that has had a tremendous impact
on your ability to execute as a team?
So we talked a bit about this already,
auto deploys and that auto merge feature.
But probably the most profound thing is
we put a little button in every single developer tool.
That's right.
It is an emoji of an octopus that is crying.
If you click it, it makes it also just type in
what's gone wrong.
And then we have our developer productivity team
read all of those and use them to prioritize
what they're up to.
So just like frictionless problem reporting
turned out to be really valuable.
We call those paper cuts.
Crying octopus.
I love that.
Yeah.
Who else from Stripe or former Stripe
should I have on the podcast is my final
lightning round question.
I see.
I see how this works.
This doesn't have to work.
Also Stripes, I think you should have on.
Emily Sands is our chief economist
and leads our information work.

I think that's data science, but also the teams
that build all of our or a lot of our internal tools.
And I think Emily just has great frameworks
for thinking about how to translate
what's really going on for users
into great sets of metrics that you can then
use to kind of get the right action happening,
as I described before.
That's super important.
And the second person would be Michelle Boo.
So Michelle joined Stripe as an engineer
in the very, very early days
and has been with us for obviously a long time.
She's really our principal product design architect
in terms of the actual abstractions
that we use to model our users' businesses on Stripe.
And I think she has very deep insight
into how to think about getting those things right.
And I think you would enjoy talking.
David, I so appreciate you making time
for this, especially during this hectic period
ahead of sessions.
Two final questions.
Where can folks find you online if they want to reach out?
Maybe learn more about what you're up to.
And then how can listeners be useful to you?
Cool. Okay.
Well, I am at DPS on Twitter.
That is definitely the best place to get hold of me.
But you can also check out my personal blog
at blog.singleton.io.
And useful.
Honestly, please send me Stripe feedback.
You can do that on Twitter.
You can discover my email address on my blog as well.
I would love to hear how we could serve you better.
David, thank you again for being here.
Thank you.
It was a lot of fun, Lenny.
Same. Bye, everyone.
Thank you so much for listening.
If you found this valuable,
you can subscribe to the show on Apple Podcasts, Spotify,

or your favorite podcast app.
Also, please consider giving us a rating
or leaving a review,
as that really helps other listeners find the podcast.
You can find all past episodes
or learn more about the show at lennyspodcast.com.
See you in the next episode.