

# Mental models for building products people love ft. Stewart Butterfield

90 MIN · YOUTUBE · [HTTPS://WWW.YOUTUBE.COM/WATCH?V=KLE-ZY5R0MK](https://www.youtube.com/watch?v=kLe-ZY5R0MK)

<https://www.youtube.com/watch?v=kLe-zy5r0Mk>

## SUMMARY

*Stewart Butterfield, co-founder of Slack and Flickr, shares insights on product development, leadership, and the importance of creating value for customers. He emphasizes the necessity of continuous improvement, the balance between friction and comprehension in user experience, and the significance of generosity in leadership.*

- Butterfield's mantra: "In the long run, the measure of our success will be the amount of value we create for customers."*
- He advocates for recognizing and addressing the owner's delusion, where creators fail to see user needs due to their own biases.*
- The concept of utility curves helps prioritize product features based on the value they provide versus the effort required.*
- He believes that friction in user experience can be beneficial if it aids comprehension, rather than simply reducing clicks or taps.*
- Generosity in leadership fosters a cooperative environment and can lead to better employee morale and customer loyalty.*
- Butterfield's experiences with Slack's product decisions highlight the importance of user empathy and understanding the emotional connection users have with software.*
- He discusses the challenges of pivoting and the importance of making rational decisions when a product isn't working.*
- His leadership style emphasizes ethical business practices and creating a positive impact on both employees and customers.*

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

>> Slack was famous for being one of the early consumerized B2B SaaS products. >> At more than one company all hands, I made everyone in the company repeat [music] this as a chant. In the long run, the measure of our success will be the amount of value that we create for customers. And you can put effort into demonstrating that you have created [music] this value and stuff like that, but there's no substitute for actually having created it.

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

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

>> You started two companies, both famously pivoted. I imagine many people come to you for advice on pivoting. >> The decision is about like, have you exhausted [music] the possibilities? Creating the distance so that you can make an intellectual, rational decision about it rather than an emotional decision is essential. And the reason I say you have to be coldly rational about it is because [music] it's [ \_ ] humiliating. Today my guest is Stewart Butterfield, a founder and product legend who rarely does podcasts. Stewart founded Flickr and then Slack, which he sold to Salesforce in one of the biggest acquisitions in tech history at the time. There is so much product and leadership wisdom locked away in his head. I feel like our conversation just scratched the surface. We chat about utility curves, something he calls the owner's delusion, a hilarious pattern he sees at companies he calls hyperrealistic work-like activities.

What he's learned about product and craft and taste and Parkinson's law, why you need to obsess with not making your users think, [music] the backstory on his legendary "We Don't Sell Saddles Here" memo, and so much more. A huge thank [music] you to Noah Weiss, Chris Cordell, Ali Rail, and Johnny Rodgers for suggesting [music] topics and questions for this conversation. This is a really special one and I really hope to have Stewart back to delve even deeper. If you enjoy this podcast, [music] don't forget to subscribe and follow it in your favorite podcasting app or YouTube. It helps tremendously. And if you become an annual subscriber of my newsletter, you get 17 incredible products for free for an entire year, including Devon, lovable Repl it Bolt, Linear, Superhuman, Dscript, Perplexity, Warp, Granola, Magic Patterns, Raycast, and Maven. Head on over to [lennysnewsletter.com](https://lennysnewsletter.com) and click product pass. With that, I bring you Stewart Butterfield after a short word from our sponsors.

Here's a puzzle for you. What do OpenAI, Cursor, Perplexity, Vercel, Plaid, and hundreds of other winning companies have in common? The answer is they're all powered by today's sponsor, WorkOS. If you're building software for enterprises, you've probably felt the pain of integrating single sign-on, SCIM, RBAC, audit logs, and other features required by big customers. WorkOS turns those deal blockers into drop-in APIs with a modern developer platform built specifically for B2B SaaS. Whether you're a seed-stage startup trying to land your first enterprise customer or a unicorn expanding globally, WorkOS is the fastest path to becoming enterprise-ready [music] and unlocking growth. They're essentially Stripe for enterprise features. Visit [workos.com](https://workos.com) to get [music] started or just hit up their Slack support where they have real engineers in there who answer your questions super fast. WorkOS allows you to build like the best with delightful APIs, comprehensive docs, and a smooth developer experience. Go to [workos.com](https://workos.com) to make your app enterprise-ready today.

This episode is brought to you by Metronome. You just launched your new shiny AI product. The new pricing page looks awesome, but behind it, >> [music] >> last-minute glue code, messy spreadsheets, and running ad-hoc queries to figure out what to build. Customers get invoices they can't understand, engineers are chasing billing

bugs, finance can't close the books. With Metronome, you hand it all off to the real-time billing infrastructure that just works. Reliable, flexible, and built to grow with you. Metronome turns raw usage events into accurate invoices, [music] gives customers bills they actually understand, and keeps every team in sync in real time. Whether you're launching usage-based pricing, managing enterprise contracts, or rolling out new AI services, Metronome does the heavy lifting so that you can focus on your product, not your billing. That's why some of the fastest-growing companies in the world, like OpenAI and Anthropic, run their billing on Metronome. Visit [metronome.com](https://metronome.com) to learn more. That's [metronome.com](https://metronome.com).

Stewart, thank you so much for being here and welcome to the podcast. Thank you for having me. I'm excited. I'm even more excited. I'm so honored to have you here. I never told you this, but you've been towards the very top of my wish list of guests to have on this podcast ever since I started this podcast a few years ago. So, I'm very excited that we're finally making this happen. I have so many questions for you.

My first question is just what the heck are you up to these days? I feel like ever since you left Slack, we haven't heard much from Stewart. I'm curious what you're up to. Hopefully you're just chilling. I'm mostly just chilling. I left Salesforce um 2 and 1/2 years ago and I have a 2 and 1/2 year old. So, she was actually born 3 days after my my last day. So, a lot of time with family and it's like an enormous privilege to be able to spend time with young kids while they're while they're young. Um no new company to announce or anything like that. Um I do get a lot of emails and texts. Like basically like every 3 to 6 weeks there's this cycle because Cal Henderson, who's the CTO of Slack, and who also we worked together on Flickr, so have worked together now for 23 years, have been talking about what we want to do next um if there is something. But you know, honestly, the the big challenge has been uh I think these things are kind of destroying the world. Um and what we're good at is making software. Um so, if you find some way to make software that helped people use their phones less often, then that would be of a a big winner, but uh haven't come up with anything good. A lot of philanthropic work. Uh nothing to announce there yet, but there's like some cool um projects I'm working on and a lot of like just personal creative art projects and supporting other artists and and stuff like that. To prep for this chat, I talked to so many people that have worked with you over the years to try to figure out what you taught them about building product, building teams, building companies that most stuck with them, that most helped them build amazing products. The first is a is a concept called utility curves. This came up a bunch across so many people that have worked with you.

Talk about what is a utility curve, how you think use that to build better products. This is pretty easy cuz it's a very familiar S-curve where, you know, you have uh it's flat, it starts arcing up, and then there's a really steep part, and then it levels off again. And on the horizontal axis, uh you can think of cost or effort. And on the vertical axis, it's value or convenience. It kind of depends exactly what you're talking about. But the idea is um the first bit of effort you put into something doesn't result in a huge amount of value, and then there's some magic threshold where it produces an enormous amount of value, and then continued investment doesn't really pay off. The most basic example I can think of is let's say you're making a hammer. And on that bottom axis, it's now quality. And if the hammer uh has a handle that breaks with any impact, then it's totally useless. And if you make it a little bit stronger, it's still pretty useless. And it's kind of like junk junk junk junk junk, okay, good, great. Then it doesn't matter anymore. If you're making an app, okay, this app's going to have users, and so let's make a users table in a database, and so far you have generated no value.

The reason I I felt like this was so important is because we would talk about like a feature. Um and usually features are thought of as as binary. Like you you either have this feature or you don't. And so, the argument I guess was, have we just not invested enough in this, or have we got all the value or convenience or, you know, quality or whatever that we could get out of this? Um and and we could have admitted the point of diminishing returns and it just doesn't matter. And I I think in many cases people will add a feature, it's not good enough, um and so people don't use it or appreciate it, but now you've added some complexity to the app, and then people give up or take it back or, you know, they tried something in in testing and they don't get the results they want, and so they decide that this is a thing isn't worth doing. And so, we would try to really investigate and and decide whether we were on the first shallow part of the curve, the second shallow part of the curve, or we were, you know, we're just coming up to it.

So, I think it's a lot easier to understand the value of this when you're talking about a specific app and a and a specific uh feature. But um I think it was ultimately helpful in getting people to like understand whether something was worth it or not. Okay, so just to mirror back what I'm hearing, there's kind of this if you visualize this curve, at the bottom it's like, "I don't even know what this is." And then up the curve is like, "Okay, I sort of get it."

And then at the top is, "Okay, I can't live without this now that I understand what this is for." It feels like it's like a really it's a different way of thinking about getting to the aha moment for someone where they see, "Okay, saved items. I get it. I need to use this constantly." So, it feels like this works both for a specific feature and also just for Slack. Like getting people to even understand, "Here's what Slack can do for you and then now I can't live without Slack. And essentially this is a lens you use to figure out where to spend product resources because if you don't get up that curve to I get it and I can't live without it, nothing else matters. Is that the way is that the framework? Yeah, yeah. And I think then you layer on another concept like the um >> [snorts] >> Bezos used the term divine discontent.

Uh the line actually moves because once people are familiar with a piece of software or the way a feature is implemented or something like that, their standards go up and so the there's like this competition. And again, this axis can be utility is the best general term for it, but it could be quality, convenience, um speed, it could be any any number of things. But as you improve your search capability or if as as you improve your login experience or your forget experience or your checkout experience or whatever, everyone else's as well and so there's this continued investment.

And when you know, forget about thinking about a new feature, you're looking at how the product works overall and usually things get kind of implemented once and then if they're lucky, they get improved upon periodically. Most things get improved upon very infrequently and some things get improved upon never. And you know, just I'm going to give an example at the absolute extreme cuz I don't actually don't know how long this has been, but I try not to criticize other people's software so much because I'm very familiar with the tradeoffs and prioritization and how hard it can be and blah blah blah blah. But okay, so it most people have the Gmail calendar app on their phone. I travel a fair bit. I'm mostly in the Eastern time zone, sometimes in Mountain time, sometimes in Pacific, sometimes in English time and sometimes in uh Japan, Central Europe.

There's like a you know, maybe 10 time zones, 12 time zones um that I would ever choose. When you uh hit the option to set the time zone on an event in Google Calendar on the iOS app, it presents all the time zones in the world in alphabetical order. And that's like the I mean, there's probably worse orderings, but that it doesn't there's no value in that. Um and even when you start searching, um it still presents them in alphabetical order by country with that term. So, if I'm in California and I'm trying to set an appointment for next week when I'm back in New York, and I type in e a s t, then I get a bunch of garbage. Okay, East ern e r n. And then the first one is Eastern Australia, New South Wales.

Um and then Eastern Australia, Queensland. And then Eastern Australia, Daylight Savings and Eastern Australia, Standard Time. And then you're like, well, [ \_ ] What I I can't remember which one is Daylight Savings and which one is Standard Time and we know what. I could keep going like this for a while. This is an app that's used by at least hundreds of millions of people, presumably every single Google employee. It's bananas how bad it is. There's so many like there's all these clever things you could do. Like you know me, I'm on the West Coast, first option should be the East Coast and vice versa. But it definitely shouldn't be that every time zone is presented with equal you know, value.

There's I don't even know, but like a couple hundred time zones. I grew up in Canada, there's a Newfoundland has its own time zone which is offset by half an hour. The population of Newfoundland is about half a million people, not that many people go to visit Newfoundland, maybe a million people in all of history. So like a million and a half out of 8 billion people. And there's Newfoundland, you know, like the same with China time which is like 25% of the world's population and this stuff.

Anyway, I and that was a little bit longer than I intended to go on this example, but it is um uh it's crazy cuz no one's going to switch to Gmail or to to G Suite Google Calendar from Outlook Exchange because the time zone picker is good. So and maybe in some sense, it doesn't matter. But at the same time, there's a real value in in delighting customers and there's an emotional connection that they form or don't form. Um And in some cases that could be really positive like they would recommend it and when they switch companies or decide to start their own company, they're going to choose to use this product or advocate for it because of that emotional connection. And vice versa, they'll also be like, I hate this thing, it drives me bananas, I really think we should stop using it or you know, advocate for the alternative.

Uh and I I think people just don't appreciate or come back to those things often enough. And then there's this category of like really essential parts of the app. Again, like account creation, sign up, forgot password, you know, things like that that for most organizations very infrequently get a lot of love and and iteration and improvement despite the fact that the kind of the quality bar has gone up across the board and it continually goes up.

>> Let's go down that rabbit hole a little bit more around delight and craft. Slack was famous for being one of the early, let's say, consumerized B2B SaaS products. Slack leaned into delight and experience and craft and a great experience. And you just as a product leader, I'd say is are known as as very taste forward, very craft oriented leader, which is pretty rare and I think continues to be rare. So, there's a few things I want to talk about

here. One is uh taste. I heard at a talk you gave at a really unique you gave a talk on taste and you have a really unique perspective on just what taste is, what product taste looks like. Can you share that?

There is a lot of, you know, going back to the utility curves again, people who are obsessed with this one little thing and you know, keep on adding more and more detail improvements beyond the point where it makes much of a difference. But um I guess a couple things about taste. So, one is can you learn to develop it? I think so because like the word literally comes from experiencing food and putting stuff in your mouth. And can people become better chefs with training? Yes, absolutely.

Undoubtedly some people have a natural advantage or or you know, born with this ability to make discernments that are difficult for other people to make and stuff like that, but you can definitely practice and you can definitely get better. The second thing I'd say is um you can create a real advantage for yourself, for your product, for your company um by leaning into it because most people don't have good taste and don't invest. And so, you're probably familiar with the again, Jeff Bezos line, your margin is my opportunity.

Um and pretty obvious what he meant by that. I would tell the story at Slack over and over again and actually made it part of the new hire welcome. So, uh I'm I'm going I'm in Vancouver at our Vancouver office and I'm going for a walk with Brandon Velestuk who's our um at the time creative director for product development. I think that was his title. And uh we're in the Yaletown neighborhood in Vancouver. So, there's like really narrow sidewalks cuz it used to be a warehouse district and now it's like, you know, fancy restaurants and nail salons and geeks and stuff.

And as it does in Vancouver, it starts to rain. We don't have umbrellas. We're walking back to the office and most people have umbrellas and we're you know, kind of on these narrow sidewalks with people coming towards us with umbrellas. And we noticed how few people would move their umbrella out of the way. And of course, you know, the other person their umbrella, the pokey bits are exactly at eye level for for people walking towards them and we would get like, you know, forced off the sidewalk or like having to duck down or whatever. And we it became a game where we were guessing, is this person going to tilt their umbrella out of the way so we can pass or not? And something like 1/3 of the people would do it. And we had this conversation about it where it's like, okay, I can think of three reasons why people wouldn't do it. One is they have very few avenues in their life to exercise power and this is one of them and they're just like want to get out there and dominate people and cause suffering. So, shouldn't ascribe to malice that which can be ascribed to ignorance. And so, that's probably you know, probably is the explanation for a tiny tiny tiny percentage of people. But the other two explanations aren't that great either.

One is that uh they see it's happening, they see they're pushing other people off the sidewalk or poking them in the eye or whatever and they're just like, [ \_ ] that's too bad. I you know, I wish there was something I could do about that, but I can't think of anything. And the last reason is they just don't notice at all. But they're just oblivious to their impact on on other people and they're they're so in their head.

And I can't really think of any other explanations for it besides that. And so we would say at Slack like tilting

your umbrella is our opportunity. You know, that's not a great rephrase of the your margin is my opportunity, but your failure to really be considerate and uh exercise this courtesy and really be empathic about other people's experience is an advantage you can create a critical advantage. And I think that there's many reasons why Slack was successful at the moment it was successful and I think we had a bunch of really wonderful tailwinds and all of that stuff, but it wouldn't have grown the way it did without those little conveniences which caused people to form an emotional connection because a lot of our growth came from you know, startup A uses Slack and then someone leaves startup A for startup B and startup B doesn't use Slack yet, and they would be like, "Oh my god, you guys, you really get This is This is so good. You've got to try it."

And uh And the the spread was driven by that cross-pollination and and people really genuinely advocating for it. That is an amazing metaphor. I I >> [laughter] >> I love that one moment became like a value of product craftsmanship, but it's like >> totally irrelevant business like is not a very common use case, and on you know, company swag and and stuff like that. Is there an example I mean I imagine there are many, but from the time of building Slack, especially in the early days, where you chose to go big on craftsmanship and experience and and delight versus speed, where you thought looking back that was a really great idea and it worth really core to success? Here's a bunch of little examples.

Um so someone else came up with this idea, and I'm I'm trying to remember who it was. Um but I it's a maybe Andrew maybe Ben Brown, something like that, who was like, "Okay, where Why do we ask people for email address and password if um they their ownership of the email address was the thing that allowed them to create the account in the first place? Why don't we just ask them for their email address and then send them a link?" And so when Slack's first version of the mobile app came out, we're like, "Typing your password on your phone, if you have any you know, minimal threshold of password hygiene is a terrible experience. They have capital H, lowercase Q, 6 correct period.

So let's just have them enter the email address. We'll send them a link. The link will automatically open the app and and authenticate them." And so there's one a little example. Wow, so you guys invented the magic link uh experience. >> someone else invented I want to be clear that. Uh I I had seen that idea somewhere else like someone else blog post about it or something like that, but we were the first ones that to my knowledge that that really kind of like scaled that and made it uh a standard.

There is another one which we we you know, we really puzzled about in the very early days where people have a long history of using messaging apps from like AOL Instant Messenger to SMS to um WhatsApp, where their expectation is they get a notification for every message that's received. And in the case of Slack, that doesn't make as much sense because you're a member of many channels, and the messages may not be for you. And so that's why we have that at tagging people and um you know, We certainly didn't invent that. That was that was Twitter.

Um but what we realized was people were signing up for Slack to you know, and it's like one engineer on this team inside of this larger organization inside of this larger company, and they would pull in the person next to them, and they would say, "Let's try it out." And then they would send a message, and then the person would be

like, "I didn't get a notification. This is this is bullshit." Um so we reluctantly decided that we had to send notifications for every single message as the default for new accounts.

But once you had I don't remember what the thresholds were. I think it's what once you had received 10 messages, we would pop up this little thing that says, "Hey, you have our default settings for notifications. We don't want Slack to be noisy for you. Would you like to switch to our recommended settings?" And then they would just click a link, and it would you know, have what should be the default, which is you only get a notification if it's a DM or someone tags you. But we realized it was worth that investment to get people over the hump. A much more uh Well, here's a I'll give you one one more simple one and and then one kind of more complex one.

People would just like the I can't remember if it's called urgent or important, but the flag in in Outlook that like you know, set the priority of a message for the recipients, always got abused inside of every company. As as soon as someone does it, like everyone's like, "Okay, I'm going to do that, too, for my message." And so all of your messages have the little flag, and it's becomes useless. We have at everyone, which causes a notification to be sent to every member of the channel when the message is sent.

And people would start you know, like someone would find this feature inside of an organization. They would at everyone. Everyone would get a notification. And then the next person to send a message was like, "Well, my thing's more important than um Bob's saying. I'm going to also at everyone." And it became really obnoxious, and people would complain about it, but it was a I don't know. I guess tragedy of the commons is not quite exactly the same thing, but it was a this real dynamic that happened over and over again. So um we came up with what was called the shouty rooster. And internally we said, "Don't be a cock." Um but we didn't obviously say that publicly. When you at everyone, a little rooster would pop up, and it would have like these sound waves coming out of its mouth and being really obnoxious, and say, "Hey, this is going to cause a notification for 147 people in eight different time zones. Are you sure you want to send this message with the at everyone?" Um and of course that worked amazingly, and it dropped off. Um and again, it was really trying to shape people's behavior so that they use We want to Slack to be very flexible, um but we knew that there was ways to use it that would be annoying and and difficult for everyone.

And so try to shape the communication culture inside the organization to take best advantage of it. That feature still exists. I see that rooster all the or no, I don't see it all. Well, actually I do at channel cuz I run a big Slack. Uh so I see that rooster and that survived. Yeah. Um yeah, that survived and and good uh because it's Sure. It was a trivially easy thing to implement, and made a really big difference. But it also taught people how the product worked cuz I mean people probably didn't know that at everyone or at channel um Right. Well, didn't think about the cost at least. Genius. Yeah.

Here's here's one more. Uh So we decided we're going to do do not disturb as a feature. Um >> [clears throat] >> And we had this not a conundrum, but we you're trying to take into account all the all the different uses of Slack cuz at the time we be implementing this like 2017, there was you know, tens of thousands of paying customers that organizations, hundreds of them, probably millions of users, maybe hundreds of thousands of

organizations. I don't remember how many. Um and everyone had set up stuff the way that they liked it, including things like ops alerts going into channels for on-call engineers for for you know, some of the biggest systems uh and and apps in the world.

And so we couldn't just like deploy it right away. We realized that some of the decision-makers, the owners of the organizations, were going to have really strong opinions about this. We also realized that some of the engineers are going to have strong opinions, and we wanted to figure out a way to kind of balance the concerns and give people appropriate means of control. So we came up with this really elaborate system for the rollout, which was we told everyone um I'm sorry, every every Slack administrator that this was coming weeks before it came.

And we told them that we were going to set a default for their organization, which I believe was um either 7:00 p.m. to 7:00 a.m. in their local time zone or or 8:00 a.m. to 8:00 p.m. I can't remember which was. But also that they could override that default. And also that the individual end users could override that system owner default. And finally that the system owner could, if they changed the default again, would override all of the the end users' preferences.

And then the end users could uh override them again. And it it wasn't to create this dynamic where people are who are at war, but so that you could change a policy, and then people could still customize and and stuff like that. But this was you know, a much longer and more convoluted process, but it allowed the millions of people who were using Slack to get the feature without creating a bunch of conflict and without people turning it off automatically. And I think critically, with setting a bunch of defaults um because if we didn't set the default, most people wouldn't turn it on at all. You know, like there wasn't If we didn't default you to do not disturb from 8:00 p.m. to 8:00 a.m., um you probably if you're the average person, wouldn't ever do it yourself. Um so that's a another elaborate example uh where I think that investment made sense because it was a critical feature for a lot of people, and um if we hadn't done it that way, I think it would have caused a lot of complaints and conflict and and stuff like that. Those are amazing examples. I very much appreciate that do not disturb feature when you guys launched that. I still remember that coming out. I'm sure a lot of people are very thankful for that. Yeah.

Something else I heard that you often espouse, which is counterintuitive to a lot of people, is is about friction, friction in the product experience. That friction is actually often a good thing. They actually it's a feature, not a bug a lot of times, if you use it well. Talk about your your experience there. Yeah, well, so yes, um and there's also another um issue around friction, which is it it became like a mantra or just like a kind of uh an assumption that you should always be trying to remove friction.

And uh in in some cases, that's true. You know, we would talk about it Slack like it it was hard to market. It's hard to explain what it was if you had never used it before. You could say a messaging app for businesses or or whatever. But you know, like a critical disadvantage to Slack doing out-of-home advertising, putting up a billboard, versus beer or cars, is no one needs to be explained why people would want a car or beer.

Um but everyone would have to be explained why they why they want Slack. Um and so the problem there is is comprehension. And this will come up uh an enormous amount. So now imagine you want to get tickets to the Taylor Swift concert in San Francisco. And you go to the Ticketmaster website. If you think about both your your comprehension, it's perfect in this case. And um that translates into the specificity of your intent. And the degree of your intent is also kind of maxed out. So like, I really want to get these tickets. And I know exactly what they are. They're Taylor Swift tickets for this date at this venue.

And so in that scenario, it doesn't really matter if Ticketmaster's website is slow. It doesn't really matter if the payments page errors out. Like you're going to persist and and get through it. So obviously they're better to reduce friction. But in some sense it doesn't It's not a huge amount of of value in doing that. Uh for most creators of of products, there are a handful of cases where that really is true for you as well. And they they include things like user registration, um authentication, uh checkout flows for e-commerce. Like I'm I am significantly more likely to buy something if there's Apple Pay or Shop Pay or something like that. I'm significantly less likely to carry through the purchase of something if the I have to manually enter all of the fields of my address one at a time rather than having one of those address pickers.

It's It's crazy, but like the issue is my intent isn't always 100%, right? And the specificity of my intent isn't always 100%. So if your thing is direct-to-consumer t-shirts and you acquire customers through Instagram ads, all of them know what t-shirts are. It's like I This looks like a good t-shirt to me. But I'm rarely like 100% intent. I might have like you know, very specific intent. My intent's like 70%. So if you're the amount of friction is uh above that, I just I'm not going to do it. But now okay, people coming to slack.com, they had some uh friend had mentioned Slack and kind of talked their ear off at some point months ago. And then they saw a news article. And then they saw someone's tweet. And then they saw an ad on a website they were visiting. And they finally said, okay, I'm going to go to this website. So their intent is like at the absolute minimum threshold. Like it's just it was before that last event happened, they were below and now they're above. But they're just above.

The specificity of their intent, like I need to get Taylor Swift concerts for this date at this venue, um is also very low cuz they're like uh it's a work thing. I'm not sure it's a spreadsheet or like a calendar or do you know exactly what it is? Um So they were coming in at you know, 0.1% over these these critical thresholds. Um >> [clears throat] >> what was the challenge? It wasn't friction, right? Because it's not like they were aiming for something and they knew what they were aiming for and they were just trying to get themselves to that point.

What we had to worry about was creating comprehension in in two senses. What is this thing? Um and what am I supposed to do next? And that creation of comprehension in the in the sense of explaining stuff, that creation of comprehension in the sense of the design of the of the UI, of the screen, of the page or whatever and the um the visual hierarchy and the affordances that are there and the and the um indication of things to interact with and and which thing should be the next thing to do and all that stuff. That becomes really critical and I think very very few people recognize that. They're like I want to get people who come to my web page to the sign-up form as quickly as possible. But if they don't know what they're signing up for, they don't know what it's going to do after, is it going to spam them? Um They don't know am I going to have to pay on the next step or or what? Then they're just going to back out.

And this was like a a lifelong battle because the remove friction um kind of orientation is so deep in people. Again, it really makes a difference in in those cases where people do have an intent and they do know what they're trying to do. Um It is a a poor approach when the challenge is really comprehension. And I think the secret is most 70%, 80% or whatever of a product design is in that comprehension step because like people if they do ever open the preferences tab and look at all the options, rarely have an idea.

And if you can't teach them, you know, or make it possible for them to discover what the capabilities are, then they're not going to take advantage of them and they're not going to get as much out of it. And I think that the trick is for most of the unique parts of any application, most of like the specific things that your app, your product, your software does are areas where the challenge is going to be comprehension instead of friction.

It really could be anything. Like Shopify, um the purpose of the of the service for its end users is generally going to be kind of clear. But most people most first-time store openers don't know that they can get reports. Or if they know that they can get reports, they don't know what kinds of reports. And if they know what kinds of reports they can get, they don't know how they can tweak them and how they can um you know, what the timing should be and which things that that are are more important to display. And I could go on and on and on and on. And people just don't recognize that. So like the I want to see if this is still true. I'm just going to open my iPhone and the clock app. And they had the most the the craziest description for alarms.

Uh okay, it's still it's a little bit different. Um But people can look at their own phone. Uh so I have It says alarms and then it says sleep and a vertical bar, wake up, and says no alarm and a button that says change. And then if you hit it, it says sleep is off. In order to automatically turn on sleep features and edit your schedule, you need to turn sleep on. So obviously but sleep was a good name for this thing if you already had a way of getting people to understand it.

If you don't, it's like ungrammatical and incomprehensible and why would you ever do it? And I got to guess this It's been like this for years. 90 plus percent and and maybe like 98% of people just do what I do, which is that you just create up like I want the alarm on and I'm going to set the time for it. And I don't know what turning sleep on does. Um But it's a just like the the lack of comprehension prevents people from getting the value. And I'm sure that there's a bunch of value behind turning sleep on, whatever that means. And people spend a lot of time on those features and it integrates with like biometrics in your watch or who knows. I I again, I still don't know because turning sleep on is like I don't What does that do? And what is it going to cost me and what impact is it going to have?

Those examples are just to me all over the place. And um the reason I don't use most software where there was an actual choice point or the reason I don't use most features where there was a choice point for me is because I didn't understand what they're going to do and I don't give a [ \_ ] And if there is one mantra that I would use to replace that, it's don't make me think. I don't know if you remember that that book.

>> Absolutely. Yeah. And honestly it's been many more than 10 years since I read it. So I don't even remember all the examples in the book. But as a mantra, that was like up there with utility curves cuz for two reasons. One

is it's just like it it's expensive to make a decision. Like you literally burn glucose. Like there's a metabolic action. There's like ATP created in the mitochondria in your neurons and like a bunch of stuff is happening. And people do get decision fatigue and there is like you know, a cognitive cost of all these things. But also there's an emotional aspect, which is if you if your software kind of stops me as I can and asks me to make a decision and I don't really understand it, you make me feel stupid. Right? I'm like, I don't understand this.

I Some people are are you know, maybe their orientation is looking that the software is stupid. But I think most people are like, oh, I'm dumb. And if you ever talk to people who aren't especially technologically savvy, you know, like the canonical example is like people who are under 50 talking to their parents about using some piece of software and what they're supposed to do, the parents almost always feel stupid. They're the ones that are that are wrong.

Um and so if you're causing people to think, in the the best case, it's like unnecessary use of their you know, biological resources. And then in the worst case, you've like now made them feel bad. Like emotionally bad. And they're going to associate that with the product forever. And I mean, these are things that are just kind of rolling one into the other. So I'm going to uh keep going with one last thing cuz they just kind of come together, which is along with reduced friction, it's like reduce the number of clicks or taps it takes for someone to accomplish something, which is almost always exactly the wrong thing. Like it's um the easiest way like you could make any action in your app a single click or tap by just exposing every single possibility on one screen that scrolls for thousands and thousands and thousands and thousands of pages. Right?

And obviously that's terrible. So why do people think that a little bit of that is good? And here's an example like You open a menu, there's 14 things that people might want to do. Um Okay, level one is group them into like items and put a vertical uh sorry, horizontal divider between them. So at least people can kind of chunk and and see what there is. Step two is present the two or three most common things or the five most common things or whatever and then have some form of other. And you you know, then you go to a sub menu that has more items. And you the decision of like how to tune that becomes incredibly important. Uh I'm going to pick on Google again just because it is I feel like I'm Donald Trump here, but I'm going to interrupt myself again with a story. It's And at some some conference or event, I don't remember what it was, and this is probably 8 years ago, and uh it's we're in the bar after the session has ended at this thing, and John Collison from Stripe is there, um and uh Sundar, CEO of Google, is there, and John I I'm sorry, Patrick goes up to the Sundar, uh and they could talk about anything, right? Like, you know, Stripe wasn't at the behemoth it was now at that point, but it was still like, you know, a significant company, it was up-and-coming.

And what does uh Patrick want to talk to Sundar about? It's in the Gmail um app, the dragging of people like you when you reply all um to a message, you often want to change so the the two recipient to the CC, and move someone from CC to two or something like that. And just how physically, like the the degree of dexterity that's required to do that inside of the Gmail app is very high. It still hasn't been fixed. But it really struck me that like, you know, Patrick could have asked for anything. It could have been any topic.

It could have been a partnership. It was like it was so irritating to him um that it worked like this, and couldn't

couldn't get it over it. So, anyway, back to bashing on Google, who in many respects do an incredible job, and there's all kinds of amazing stuff they do, and blah blah blah. But, the the Gmail actions on an individual email are broken into two very long menu items that are different. And one of them doesn't exist on either menu. There is an unlabeled icon is the only way to do it, and that's the to mark something as unread once it's read.

Um I have no idea why some of the actions are in one menu, and some of the actions are in another menu. I I think it's because some of them have to do with the individual email, some of them have to do with the whole thread, but it doesn't seem very consistent. Every possible thing is listed there in one place, and so it becomes incredibly difficult to use because sometimes you have to tap and bur Tell me what it means read all of the options and say, "Okay, that I've used the process of elimination, and it's not here, so it must be it must be there."

The Uber doesn't work like this anymore, but when I first brought this up to people inside of Slack, there was a moment when the Uber app, when you opened it, was just "Where would you like to go?" and other. And other was everything. Like, change your payment method, uh set your location for your vehicle, any anything you could do in Uber. And that was perfect because almost all of the time people just wanted to choose where they wanted to go. Sometimes you wanted to change where your pick up was cuz you weren't there yet, or or whatever. And that was just like, what could be simpler than I'm going to tell you where I want to go, or I'm going to choose something else. I really tried to push people to what is the thing that people, or what is the two things, what is it, maybe three things that people could want to do here, and then put everything behind other, and then um if it takes them eight clicks or taps to do something, but every single one is trivially easy, that's great. If it, you know, you reduce that to two clicks or taps, but every part of it is this fraught decision where I'm opening all of the menus and trying to figure out like which thing is the right thing, and and like, the more comparing three things to each other is this difficult, four things it's kind of like geometrically more expensive to compare 15 different options all to the other to see if this is the one that you might want. Um they you know, just becomes impossibly expensive. So, to me those are all really um connected, and if we if people could get over the idea of reducing friction as a number one goal, or reducing the number of clicks or taps to do something, and instead focus on how can I make this simple? How how do I prevent people from having to think in order to use my software? How can I make this trivially easy? One one last example uh because this this was what really influential for me. So, I was going back and forth between Vancouver and San Francisco at the time when we were talking about all this inside of Slack, and I was behind a teenager in line to board the plane, and it was like, you know, we're on the jet way, it took a long time. And I was watching her use Snapchat, and it was insane. Like, she was tapping at least four times a second, sometimes like six or seven times a second. It was like dismissing stories and doing stuff, but there was a fluidity to it because everything was like a dent. Do I want to see this again? Do I want to see the next story from this person? Do I want to ask her to a different person? Do I want to like instead she she um a notification came up, she answered someone's thing, she took a selfie of herself. And everything was just like So, she was you know, tapping four times a second for 6 minutes. I mean, probably there were some some breaks in there. And that was like the highest and best use of Snapchat for a 15-year-old girl in 2016 or whatever that was.

Um And imagine if the goal was to try to make her tap less. You know, like how how much of an impediment it would have been to the experience that both her and Snapchat wanted to create. It's so fun to listen to this and the examples you gave of it gives us a lot of insight into the way your mind works of just constantly unsatisfied

with the way other products work, with your products. And I think that's core, like Patrick is a good example of Stripe. I feel like that's a recurring theme with very successful product leaders is just constantly unsatisfied and unhappy with how things work.

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

Yeah, I I think um in my more cynical examples, I would I would say to people like, "Stop Stop what you're doing for a second. Close your eyes, take a couple deep breaths, and then pretend that you're a natural human being, and open your eyes again, and then look at this thing and see can you figure out what it's supposed to do or say, or what your what action you're supposed to take, or what the impact will be if you take that action?"

There's a whole another related cycle, but before I get into it cuz I know that I'm verbose, um I want to wrap up your your last example of people being unsatisfied. So, here's the quote that I was trying to find. This is um >> [snorts] >> 2014, so like that that was the year that Slack actually launched officially uh in February, and this is now like near the end of the year. I was interviewed by uh MIT Technology Review and asked if we were working to improve Slack.

Uh I said, "Oh god, yeah. I try to instill this into the rest of the team, but certainly I feel like what we have right now is just a giant piece of [ \_ ] Like, it's just terrible, and we should be humiliated that we offered this to the public." Not everyone finds that motivational, though. So, I came into the office the next day, and people had printed out on like 40 pieces of 8 and 1/2 by 11 paper that quote, and like and pasted it up on the wall. But to me that was like, "You should be embarrassed by it." Like, it should be a perpetual desire to improve. You shouldn't have to be like, "Oh, this is great." And you could be proud of individual pieces of work, but in the aggregate, if you can't see almost limitless opportunities to improve, then you shouldn't be designing the product, or you shouldn't be in charge of the company, or you shouldn't um almost nothing, you know, again, you could reduce it down to a tiny feature, is anywhere close to to perfect. And if A, that's acknowledged freely inside the organization, and B, people think about like continually improving as the goal, and that could be like Six Sigma, Toyota Kaizen, like that kind of side of thing, or it could be that story that um I can't remember his name right now, but the guy who started Bridgewater tells about um Ray Dalio. Yeah, Ray Dalio. Uh in his book talks about um Michael Jordan learning to ski. Every time he messed up, he wanted the ski instructor to tell him exactly what he was doing wrong because to him every one of those was like a gem that he could collect, and and he could, you know, actually become a good skier, and what he wanted to do was become a good skier.

That requires a lot of trust inside the organization. Um but if you can get to the point where like, "Hey, we're all trying to to find improvements, we're trying to be critical of each other, we're trying to make this as great as it can possibly be." Um Not always, not with every person, but most of the time with most people you can get them to the point where that like really direct criticism is actually motivational. It is like, you know, people are uh grateful to have like the feedback, whether that's coming from their peers inside the company or from end

users of the product cuz you realize, "Oh yeah, that is that is bad, and we should fix it."

This episode is brought to you by Lovable. Not only are they the fastest growing company in history. I use it regularly, and I could not recommend it more highly. If you've ever had an idea for an app, but didn't know where to start, Lovable is for you. Lovable lets you build working apps and websites by simply chatting with AI. Then you can customize it, add automations, and deploy it to a live domain. It's perfect for marketers spinning up tools, product managers prototyping new ideas, and founders launching their next business.

Unlike no-code tools, Lovable isn't about static pages. It builds full apps with real functionality, and it's fast. What used to take weeks, months, or years, you can now do over a weekend. So, if you've been sitting on an idea, now is [music] the time to bring it to life. Get started for free at [lovable.dev](https://lovable.dev). That's [lovable.dev](https://lovable.dev). This makes me think about a rant that you have about how it takes a lot of work to make anything work at all. That just the default state is not working. You just share what you share there.

Yeah. Um I mean, so this is a lot to do with um And maybe this is more recent than other shows I've been in politics a lot for a um But uh By the way, if any of your Anyone listening to this can help me find this tweet storm from somewhere between 2016 and 2020. I don't have a precise idea, and it was this guy's thread about how hard it was to get um a stop sign set up. And I think I believe it was in response to someone claiming that Bitcoin is going to replace US dollars. Uh It was something something about crypto.

And his point was like, "Here's what happened when we tried to get a stop sign put up on a residential street in my neighborhood." And the literal years it took and the number of agencies that were involved, like the engineering department, traffic planners, the HOA, and the I don't remember all of the organizations um because I think if I did, then I could search better and find this again cuz it was truly a masterpiece of how difficult it is to get a stop sign put up in in most places.

The message that I hear from most politicians, and unfortunately this works really well, is you know, things should be good, but they're not because someone is doing something bad, which is preventing the goodness. So, um billionaires are making things unaffordable, or immigrants are taking your jobs, or um lazy freeloaders are sucking off the government teat and causing us all have to pay more taxes or something like that. The reality is like almost nothing works. It's It's actually another call I sent This case John has a great encapsulation of this, and I'm sure you're familiar with it, like that It ends with the world is a museum of passion projects because for anything to get done at all requires like not just the resources and and effort you know, required to instantiate that thing in the in the real world, but all of the politicking and the sociology and the convincing and um There's a book called Why Nothing Works recently, which is like It's not an I'm sorry to the author if they They doubt they're listening, but just give It's not like an amazingly written book. I I found it like um a little bit repetitive, uh but the the content was really incredible. It's just explaining why it's so hard and how there's this progressive increase in the number of of vetoes that are available for any kind of course of action and how difficult it is. And this shows up in like, you know, in permitting for new construction and stuff like that. But it also shows up obviously inside of organizations. And the challenge is that people A, I think this is evolutionary, biological.

Uh it's hard for us to understand the world except by anthropomorphizing it. And so it like if it didn't rain this year, it's because a god is mad, and probably because we didn't sacrifice enough goats something last last year. It's hard for people to understand just that well, weather is incredibly complex and chaotic and ecosystems and climatology and all that. Same thing with the world. Like if I uh am struggling to uh you know, pay all of my bills and be able to afford me a little bit of luxury in the sense of like a vacation or a present for my kids or whatever, it's got to be somebody's fault. Like there has to be a decision that's made somewhere.

And the reality is everything is so complicated. Everything is so multivariate. Uh it's not satisfying. It's a terrible political message. Um it's much easier to say that there is like uh Well, we understand why things are bad in the way that you're concerned about, and it's turns out that it's someone's decision. Um and because of them, it's bad. And it so if we got rid of them or you know, were able to overcome their decision, overturn it, um and institute our own thing, then things would be good for you.

Uh And this really to me shows up inside of organizations as well. You know, I'll pause there. Uh I know kind of along those lines you're a big uh believer in something called Parkinson's Law. Yeah, so that the original of that is I think it's 1956. It's an article in The Economist by Parkinson. Um And uh the maxim is work expands to fill the time available for its completion. And the way that shows up This is a little bit subtle. So, like um one of the things I found since I don't have a job is there's much less time pressure, and and that maxim like uh if you want something done, give it to a busy person, the inverse is also true that like if you're not that busy, wow, basic things take a really long time. And so Parkinson actually starts off with his example of like, you know, writing and posting a letter.

Um and I I don't remember who he used for the first example, but someone who's like, you know, incredibly busy and has all these things they have to respond to. And then another case, like a retired woman who has all the time in the world, and it takes her a long time to write the letter. It takes her a long time to put it in the envelope and then go to the post office and post it.

But the real meat of it is um for me later, when he talks about the size of the organization. He uses a bunch of examples. This is again 1950s, so uh and he's British. Um So, he's looking at the the Royal Navy, and specifically he's looking at a chart that shows the relationship between the number of capital ships in the Navy, the number of sailors, uh and the number of administrators. And very familiar graph for for people looking at like any part of government, any part of like the relationship between the number of administrators at a university and the number of students and faculty, the teaching faculty. Where it's like, okay, the number of ships goes like this, and the number of sailors is like right along with it, and the number of administrators goes like this.

And the reason this this ties into the work expands to fill the time available for its completion is um people hire, and they train, and uh Here's the kind of sad truth for anyone running a company is there are exceptions. There's like certain types of engineers that are exceptional to this, but the overwhelming majority of people you hire want to hire more people who report to them. And it's not because they're evil.

It's not because they're stupid. In fact, they're smart because everyone knows that the number of people report

to you correlates with like your career trajectory, your the the amount of money that you're paid, the the amount of authority you have inside the organization, and on and on and on. So, like we would hire 27 year old product managers in Slack who immediately want to hire someone. It's like, "What the hell What would that person do?" And they're like, They articulate it this way, but essentially it's like, "Well, that person would do the product management, and then I would do strategy."

Classic. It's It's really I think the essential thing to understand about this is it's not because people are evil, and it's not because they're stupid. And it's to me very related to the everything is complex. Um and uh if you I maybe this is my Butterfield's Law. I haven't thought about this before, but I I tweeted this a very very long time ago. Like if you um Everything is simple if you have no idea what you're talking about.

Um So, the the other side of that is like if something seems simple, probably you don't understand it. And you know, there's obvious exceptions to that, but for anything that involves a large organization or a lot of human beings, if the problem seems simple, you don't get it. So, every budget process no head of engineering, no head of sales, no CFO, no GC is ever going to come back and say like, "Oh, I actually think like next year we can just hire fewer people, or we're going to keep it flat, or we're going to like shrink through attrition cuz we don't need any more people to do what we're doing." Not because they're evil, not because they're stupid. Uh but it's an almost overpowering impulse inside the organization that often leads to disastrous results. And so, there's a I'll give one example from Slack's history, and I you know, I have tried in the past to disguise this example so that no one feels bad about it, but I Fortunately, the specifics are so important to the example that it's not disguised, and so I'll just reiterate that the people involved aren't stupid or or evil. Um and one example that's um This is from the outside. So, the example inside of Slack was um We introduced threads, which is the ability to reply to a message inside of a channel. And let's say you, Lenny, post a message. I, Stewart, replied to it. You will automatically get a notification. And now Sarah, later on, replies to the same message. Both you and I, um as people who have posted in that thread, will receive a notification that there's been more activity. And so what?

So, like, you know, every single time anyone replies to it. So, when the feature first was um released or like when we did the final product review before it was released, the input box was pre-populated with at the person before you in the thread. And I you know, I was using the feature and I would like put the insertion point there, select all, delete, and then start writing my message. And it's even if I wanted to at someone specifically, I almost never remember to start my sentence with at I cuz it just made it hard to you know, reference what they were saying before.

So, I said get rid of this because A, I think most people won't use it or if they uh if they did want to at someone, they're not going to want to do it at the beginning of the sentence. And by the way, you're teaching them to use the product wrong because it's important that everyone understand that every previous poster in this thread will automatically receive a notification unless they've muted it. So, okay, we release it, 6 months goes by, and suddenly the at thing comes back. And so, I messaged someone on the team and I said, "Hey, there's been a regression at this super weird I don't know what happened, but like the at thing came back." And they said, "Oh, no, this is on purpose."

We did a bunch of research." And so, I was like, "What?" And I went through this and it was if I recall correctly, it wasn't even like P95 certainty on this analysis, but it was something like when we do this, threads are 2.17 messages long versus 2.14 messages long on average for when we don't do it. And so, first of all, why is a longer thread better? Like maybe a maybe a shorter thread is better. Like it's fewer messages that people have to go back and forth.

Also, that's such a tiny difference. Also, again, I don't remember the actual statistical analysis, so I'm not going to claim that it was um in correct me, I pretty sure this was outside the the bounds of certainty that they could have. But the real thing was Oh my god. So, you guys put flags in the product, you AB tested it, you did the instrumentation, you created tables in the in the database or whatever you're using to record all of that, you wrote queries to pull that, you created charts based on that data, you had meetings to discuss it.

And just like kind of unpacking all of the things that would have had to happen for this to come back. And it's like, you know, thousands of person hours kind of at a minimum cuz any feature change at at that scale decision is involving like a dozen people, engineering, QA, uh analytics teams, product managers, user research, and stuff like that. The problem with that was So, I think it was a bad idea, right? But the problem with that was the difference that you could possibly achieve between having this feature and not having this feature is like this much, whatever units you want. The cost of doing the analysis was this much. So, it's guaranteed to be a loser.

Like there's just there's no world in which anyone could imagine putting the at previous respondent in the thread at the beginning of the message could possibly make that much of a difference to the quality of Slack and how how much utility it provides to people and and all of that. But you know that to to like put the feature flags in, to ship new versions of the product, to put the instrumentation in, to have it all the API calls to record every action that people take, to to do all the analytics, to create the dashboard, to put that, you know, paste the screenshot of that into a Google Slides presentation, to send the invitations to the meeting, to schedule the meeting cuz someone couldn't make it, to have everyone sit down and look at the thing.

Like, you know, guaranteed loser. And I know that you that Farid told you to ask me about this um hyperrealistic work-like activities. And so, here's my my my grand theory. Um hyperrealistic work-like activities is goes along with this other concept called known valuable work to do. And when I say known, I mean both you know what it is and you know that it's valuable. And the problem with almost every organization at the very beginning, you have an enormous amount of work that you know what to do and you know that it's going to be valuable. So, like starting a business, open a bank account because like there's almost infinite generative value of opening a bank account. You you have to do it. You it's very simple to do. And so, at the very beginning of of any startup, they're like, "I'm like creating a user's table and I'm like doing salting passwords and like you're doing all the things that are kind of absolutely necessary and everyone knows exactly what they are. And so, that's like everyone's going to work in the morning and they're like, "Right on."

And like I have 10 things to do and every single one of them is like something I know how to do and it's like definitely going to be valuable. Time goes on and the relationship between the supply of work to do and the demand for doing work just starts to change. More and more people get hired. Every product manager wants to

hire a junior product manager. Every new person, you know, the first person you bring in on the risk and compliance team is like, "Oh my god, we have there's so many risks and things we have to be compliant with. We better hire more people on my team to do more risk and compliance work." Which probably, you know, to some degree is right, but we're going to have more and more of those people and they're going to call meetings with each other.

Um and now suddenly you have all these people with work to do and you've done all the easy, obvious stuff. And now your questions are like, "God, should we do FedRAMP High and make a GovSlack version?" Which is going to require us to have wholly separate physical infrastructure for the hardware that runs the the software and also a whole different operations team which has only US citizens on it. What is the possible number of of dollars that we could make from doing this and how much complexity is going to be when we want to do updates to the software? Should we update two totally separate, independent systems? And we're it just gets out of whack. And so, people end up like if you hire 17 product marketers, you're going to have 17 product marketers worth of demand for work to do. And if you don't have sufficient supply of product marketing work to do, they're just going to do other stuff. Again, very important, not because they're stupid, not because they're evil, but because they're like, "I'm a product marketer and I want to like be recognized for my work and my spouse is like has um uh criticized me because they think like I should have already got promoted in the last cycle and I really got to demonstrate some wins here and whatever it is." And so, people are like calling meetings with their colleagues to preview the deck that they're going to show in the big meeting to get feedback on whether they should like improve some of the slides.

And that hyperrealistic work-like activity is superficially identical to work. Like we are sitting in a conference room and you are there's something being projected up there and we're all talking about it. And that's exactly what work is. Now, hopefully not all of work in for everyone inside of your company, but you know, that's uh exactly what we do when we're working, but this is actually a fake bit of work and it's so subtle that I I'll do it.

You know, our board members will do it. Every exact will do it. And the further you are from like having all of the context and all of the information and the decision-making authority and stuff like that, the easier it is to get trapped in this stuff and people will just perform enormous amounts of hyperrealistic work-like activities and have no idea that that's what they're doing. And the you know, the result of that I guess is that if you are a leader, if you're a manager, director, an executive, you're the CEO, it's on you to ensure that there is sufficient supply of known valuable to do.

Um and there almost always is, but you know, it's creating the clarity around that, creating the alignment, making sure everyone understands it that that's what they're supposed to be doing. And and then obviously doing it. Amazing. I could listen to uh Stewart Rants all day. Hyperrealistic work-like activities. We need a coin this. Uh and the >> it doesn't make a good acronym. It's pretty ugly. So, we're not even going to try. And just to close the loop on that, the the solution is uh the leader recognizing this is happening and stopping it. Telling people, "Why are we spending time on this thing that is not going to get us anywhere?" Yeah. And and and that what you just said probably isn't the best way because that sounds like they're you know, you're chiding them and they're dumb when it's actually your responsibility to make sure that there's, you know, sufficient clarity

around what the priorities are and you know, explicitly saying no to things up front and stuff like that. Rather than being rude and saying like, "Hey, you guys are bunch of idiots wasting your time on this thing that doesn't matter."

Whose fault is it? You know, it's it's the manager's fault, it's the VP of whatever's fault, it's the CX whatever. It's the Hopefully it's like it's the the leader of the organization that has the responsibility to make sure that there is sufficient known valuable work to do. Um and this that's actually harder than it might appear. Okay. Uh before we run out of time, I want to touch on two other topics. One is when people think of Stewart Butterfield, I think a lot of people think of we don't sell saddles here.

Your legendary medium post that uh is just I don't know, it's become a historic piece of literature in the in the annals of product building and in startups. I haven't heard people ask you much about this recently, so let me just ask a couple questions. One is just what was the what was the reason you put that out? Why what what was the backstory on writing that memo? Why was it necessary? Well, it really was an internal memo and as a bit of a digression, um one of the the crappy things about Slack um is if all your corporate communication is on email, depending on exactly how it works and what system you use, you probably walk away with an archive of everything you said at company X. Um if it's Slack, once you're turned off, you like you lose access to all that history. And so it's kind of like, "Oh man, if I had only exported all of my messages before I left, I would have all this stuff." But that was it was absolutely verbatim. I did not change a word of what I said inside the company. When I think we were still eight people, maybe you know, at most 10, but I I think it was eight people. It was before Slack launched even. Yeah, it was before Slack launched. It was like when we were doing a private beta and the point of it was to like to start to instill those ideas as early as possible and really create this alignment inside of that small team so that it could process to survive as we grew and and scaled. Um and yeah, that that was the that was the idea. And the gist just for people that aren't super familiar with it, but we'll link to it is just it's not enough just to build a great product. You just as much have to put it effort into communicating what this does for them, the problem this is solving for them, the outcome this is going to achieve for them. Is that a good way to think about it? Yeah, and again, you know, like comparing it to beer or cars, um beer is goes back to pre-civilization. Cars are obviously invented and but you know, so at some point you had to convince people why they would want a car instead of a horse. Um for your new um AI-based recruiting tool or your calendar app or whatever, there's some reason why people you think that people should use yours instead of the thing that they're using now, which might be like a wholesale one-for-one replacement or more often is like a change in the way that you're working that has a bunch of other adjacencies and you want to expand into these other categories and while you're not just responsible for creating the product, but also to a certain degree creating the market and and creating you know, this is book positioning um which is an absolute classic. It's very short. I would recommend everyone read it.

Where the the point of it is from my perspective it's almost impossible to create a net new idea in someone's head. It's much easier to take a couple of existing ideas and and put them together. So it's much easier to say it's like Jaws meets Star Wars Star Wars or it's Uber for pets or something like that than to come up with like an actual new idea, but you you have to do that cuz you like if your thing is different in any significant way from the alternatives, um you're not just creating the product, you're creating the market. They're really kind of one in the same. The reason I wanted to touch on it is I think still people continue to not listen to this advice and can

continue to over-invest in more features, more more products, things like that. Just the the specific example of we don't sell saddles here just to quickly communicate this to folks and correct me if I'm missing anything is just instead of, "Hey, look at this amazing saddle we bought." what you want to communicate is here, "Go horseback riding. Look at this incredible experience you can have." and and then they decide, "Oh [ \_\_ ] I need to go buy a saddle to do that."

>> Yeah. And and 100% um that aspect of it is is not original cuz I think that's something that marketers have done for a long time, surely in the in marcom and and advertising. Like if you want to sell Harley-Davidsons, there are people who are going to geek out on the engines and stuff like that and like the quality of the leather and stuff like that. But what you're selling is not the motorcycle, you're selling like the open road and freedom and the wind in your hair. And if you're Lululemon, you are obviously selling yoga pants, but you're also selling like health and aspiration and being the best version of yourself and you know, a bunch of other stuff. So selling that.

Oh my god, I forgot the classic version of it. You know, like There's the ship >> There's the instead of the the screwdriver Oh. Oh yeah, the nail. Yeah, the nail. Anyway. Yeah, what is that one? >> [laughter] >> There's the Well, there's the one I think about is instead of trying to convince men to build a ship, uh instill a yearning for the sea. Yes, exactly. You know, it's a something that goes back in in history. Mhm. Okay, let me ask you about pivoting. Uh you are potentially the king of pivots. You started two companies, both famously pivoted, both from video games, which is why I asked you about that at the beginning. Into very successful companies.

I imagine many people come to you for advice on pivoting. I I let me just ask when folks come to you asking, "Should I stick with my idea? Should I pivot?" What sort of advice do you find most helps them? Yeah, I mean, I think it's um partly uh it an intuition because like obviously the decision is about like have you exhausted of the possibilities? Um and in the case where you know, we were working on Glitch, this game where we used IRC for internal communication and we added a bunch of IRC, which became the proto-Slack. And I think Slack had an enormous advantage in the fact that we were working on this for several years without actually explicitly working on it and only doing the minimum number of features that were you know, absolutely guaranteed to be successful in the sense that it was so irritating that we couldn't stand it anymore or such an obvious improvement that we couldn't help but take advantage of it.

We still had a like \$9 million left and everyone still liked the game and we were all happy working on it, but it would I think by that point I had exhausted every non-ridiculous long shot idea to make it commercially successful. And so decided to um to abandon it. But, you know, the default advice for anyone in anything is persevere. It's like you know, kitten hanging off the branch and the poster says, "Hang in there." And you know, and I think there's a idea that you just like And then there's so many stories of like so-and-so started out going door-to-door and was rejected by everyone and then suddenly there was Nike or or something like that. And like just if you stick out stick it with it long enough, you'll eventually be successful. I think you have to really be coldly rational and Now you know, some of this shows up in the book um thinking about Some of it's in Andy second book.

The title of which I'm forgetting right now, but I'm someone will know it. Yeah, Thinking Backwards and then yeah, it was the last second one I forget. Uh she actually uses Glitch and and Slack as an example of like a smart fold basically. Like I my expected value here has diminished to the point where this alternative looks more attractive. And the reason is say you have to be coldly rational about it is because it's [ \_ ] humiliating. You know, like like we I convinced so many and you have to convince so many people to get a company off the ground. You have to go to investors, you have to go to early employees and say like you should leave your other job and come work for this because you know, here's the the incredible future we're imagining. You have to go to the press and you have to make all these promises and you have users and you feel like committed things to the users and you've convinced them to give up their time for this thing.

And so it's I I think for a lot of people it feels better to just keep doing it until it dies um of suffocation due to lack of capital or something like that uh than to to admit like, "Okay, I was wrong. This didn't this didn't work." And it's it's humiliating, it's painful, it's like it's wrenching. It has a bad impact. You know, like there's when we shut down Glitch, there's a lot of people who loved it and who spent all of their free time and couldn't wait to get home from work to go play it more and that was their community and that community just like disappeared. All these people and all these identities have been created. And obviously people lost their jobs and and people who had like moved their families to a different city in order to take this job now weren't going to have a job anymore. So pivots aren't something I take lightly and you shouldn't be um I think it's very different to be like there's three of us and we started making this app and then we pivoted to a different app. That doesn't really count. You know, like if you're six months into something, you're just you're still messing around. You're trying to figure out what it is that you're you're building. It's not really a pivot.

Obviously, in this case it worked out great and there's survivorship bias and it doesn't mean that everyone should pivot all the time. Um but I think it is creating the distance so that you can make an intellectual rational decision about it rather than an emotional decision is is essential. I love all three pieces of advice of just exhaust. Once you've exhausted all the ideas, that's a really good time to see what else is out there. Yeah, exhaust all the all the good ideas. All the good ideas. All the realistic, yeah.

Yeah. The the point you made about just kind of persevering, I just had Melanie Perkins, CEO of Canva on the podcast. They went through a hundred investors rejected her before somebody finally decided to invest and she just kept pushing. Yeah, but I think that's a slightly different example, right? Cuz like the she believed in the Yeah, then she then she believed in the in the concept of the in the product and like in the in the vision. They were just trying to figure out the right articulation to get investors who ended up being obviously very very happy. Yeah, extremely happy. Oh jeez.

Okay, maybe a final topic depending on how time goes. I want to talk about generosity. I talked to a bunch of people, as I said, that have worked with you and the number one theme that came up again and again and again when I asked them about you and what they what has stuck most with them is is just generosity. So I'm going to read a few examples that I heard from folks that are examples of your generosity over the years.

So one person shared that he needed a little money before Christmas and you he said, "Stewart literally walked

me out of the building, went to the cash machine, handed me \$500, told me to go home to my family." Other folks shared that when you talked about Glitch just recently when you had to lay people off that you cried real tears when you were laying people off and then you spent an incredible amount of time helping them find new jobs and extending their severance pay, and just taking it extremely extremely seriously, much more than I think most people feel like CEOs do.

Someone else shared that you paid 100% of employees' health insurance to give them just fewer things to think about. Uh when you went public, you did a you basically created the best possible situation for employees. No lockup, direct listing. Also, with the structure of the Slack deal, people said the acquisition is very employee-friendly. There's also just a bunch of ex- That's employees. There's also just the way you thought about customers. A few examples. You gave free credits to businesses who were struggling to pay the bills during COVID. You released this fair billing, which I think was very innovative at the time, where you didn't charge you stopped charging people for seats they weren't using, even though they signed a deal to charge for those seats. A lot of times just you slipped release schedules because you just wanted to make features better and better for people.

So and I'll end with this quote. Stewart is a leader who takes the responsibility he feels for his employees personally, and to which he extends the most generous circumstances he can muster. That feels worth celebrating. So first of all, let me I just want to celebrate you. I think it's really rare and inspiring to meet a leader like that. Clearly, you've had a lot of impact on a lot of people. I don't know exactly the question I want to ask, but I guess was this is this like an In what part is this intentional? Just like this is how we win. I'm going to be very generous and help people because I know this will help long term. How much of this is just innate? This is just the way you are as a person.

I I think a lot of it is just the way I am as a person, and uh I had wonderful parents who raised me right. But I think there's also um there is a little bit of a lesson there, and I'm just going to assume people's familiarity with the the prisoner's dilemma. Um the you know, acts of generosity to me are a way of demonstrating that I am going to cooperate as we iterate in this game.

Um and if you do that, then people will also cooperate, and you both benefit. Whereas if you never really know if the other person is going to defect at the first opportunity, um then your best bet is to defect. And so, you know, there is a game theoretic aspect. Usually in games that are much much much more complicated than uh the prisoner's dilemma. Um I think another thing you know, one thing we uh I didn't touch on before, but to me it was important enough that at more than one company all-hands, I made everyone in the company like repeat this as a chant. Um it was in the long run, the measure of of our success will be the amount of value that we create for customers.

Um and then I wanted to be super clear and explicit about that because it should be if anything you're doing feels like a little bit shady, a little bit cheating, a little bit like um maximizing um in the in the at the wrong moment, or taking advantage of a customer, or anything like that, we definitely shouldn't do it because to me it was just like I mean I think it's uh literally true, but it's also like an uh an ethical way to run a business, and and

um it's not just that the ethics are good. It's like there's advantages for you. Like you're able to attract a better class of employee. Like if all your employees are are ethical, then it's going to be a better place for everyone to work, and you're going to be happier, and you're going to have fewer internal problems, and and all that stuff.

Um but I think it really is true that there's no like especially in the long run, you can't like destroy value for your customers and expect to be successful. You have to actually make their lives better. Um and you can put effort into like pointing it out to them, and and demonstrating that you have created this value, and stuff like that. But there's no substitute for actually having created it. And I think that um is incredibly important, and that implies a real generosity, whether that's in negotiating terms with an an enterprise deal, or that's like policy decisions.

You sometimes So one one time that it blew up in our face was our SLA was like for any downtime, you get 100 times your money back. Um which was like cuz it from my perspective, it's like you know, if we're down for 2 minutes, it doesn't it's like pennies. It doesn't really make any difference. If we're down for like 10 hours or something like that, then we have bigger problems than paying back people. Okay.

Okay. Fast forward, we you know, we now have hundreds of millions of dollars in revenue, and we've gone public. And like shortly after we go public, we have like one of the biggest outages we've ever had. I don't remember how long it was, but it was like many hours. But by the time we got to that scale, 100 times the money back for like the third of a day that we were down was like \$8 million or some something like that, which had to be um it didn't cost us any money because we just gave it to people as forward credits, but it meant that a bunch of revenue that we had already anticipated for the next quarter wasn't going to show up because people's credits were going to offset what they would have otherwise paid us.

And so, uh we we definitely changed the terms of service after that because being a public company is a little bit different. But uh you know, in every other respect, I think you know, they were all really important decisions that were helpful in um in us becoming successful. And was that policy was automatic? Like you didn't even have to claim it. It was just automatically get this credit. >> And and the default is you don't have to pay if you let us know. And this was we will automatically like proactively, preemptively, without any input from you, apply this credit to your account, and just like send you a message that it happened. And by the way, we will do it on the aggregate for downtime, even if the issue didn't affect you as a as a customer.

>> [laughter] >> Yeah. What was that mantra again that you uh had the company chant? I think this is a really nice way to end it. In the long run, the measure of our success will be the amount of value we create for customers. Incredible. I'm just trying to picture the entire team at Slack Hundreds of people. It felt very like Kim Jong or like Stalin or something like that. >> [laughter] >> Well, on that note, most people don't know this about you, but your uh your actual name when you were born is not Stewart. It was Dharma. Yeah. And this all makes sense as you learn that.

Yeah, it's like I had My name is Dharma Jeremy Butterfield. So my parents named me, and when I was 12, I changed it because I just like really wanted to be normal, and for some reason I thought Stewart was a normal name. And by the way, you'll notice this now that I said it. Any character except for Stuart Little, the mouse, um

anytime you see a character in in a movie, a novel, TV show, whatever, there's only the loser Stewart, and the [ \_ ] Stewart. It's it's like it's obviously in the collective consciousness, a terrible name, and I shouldn't have chosen it, and I regret it. But by the time I realized that Dharma and Greg had already come out, it would have seemed like I was like, you know, bandwagon jumping, and people thought it was a girl's name, even though in India, it's obviously only a boy's name. I'm going to add just one last little tidbit because I forgot about this earlier on, and I think it helps tie things together.

And it's called the owner's delusion. And this is basically something I posted on Twitter. The person who came up with the name later deleted their account, and so I have no idea who it was or who to credit for this. But what I had posted was and this is a long time ago, before when restaurant websites have gotten better, and it doesn't really matter because Google Local has taken over everything. But this is like let's say 10 years ago.

There's five things you could possibly want when you go to a restaurant's website, and is their street address, their phone number, the menu, the hours of operation. Oh my god, I'm forgetting the fifth thing. Oh, and and making a reservation. How to make a reservation. Um and again, this problem has like to some extent taken care of. It's actually really improved. But what you would get was like this super slow loading photo, the Ken Burns effect as it and like and then like fading in, and then some music starts playing, and then if they show you the phone number, it's not uh clickable, and it turns out it's not even text that you can copy because it's yes, it's an image. Um and they don't have the hours, they don't put the address or whatever. And it's just like, I mean what?

For sure, whoever made this website for the restaurant owner, and the restaurant owner themselves have definitely been in the position where they went to somebody else's restaurant website because they wanted to get the address or the opening hours or the phone number or whatever. So how why does it end up like this? And what should we call this? And the whoever replied to the tweet, um she said we should call it owner's delusion. And I was like, oh my god, that's perfect.

And I think that um is incredibly powerful, and it what ends up with the result like Apple naming whatever that feature is called sleep, which is like it's too hard to understand what could that can possibly mean. And that's why like people anticipate um despite the fact that when they get to their web your website for the first time, their their intent is absolutely the minimum number of micro points above the threshold required for them to actually take that action. You're like, all right. Like welcome to my website, and there's >> [laughter] >> there's a bunch of like BS, and there's a bunch of like stuff that doesn't make any sense, and the buttons aren't clickable. It's unclear what to do next because I think that my thing is so important, and I don't recognize that you are at work, and you were late this morning, and you have to go to the bathroom, and you're just like you're just like a regular human being who has like stuff going on. You're concerned that your kids will [ \_ ] up, and they're getting in trouble at school, and stuff like that. They don't they're not like subjects who paid money to go to your play, and are sitting in the audience and waiting for the curtain to go up. They're like people who are going to bounce in a fraction of a second. And so everyone should always be conscious of the audience's illusion. I love that. What's the solution? Is it have other people look at and give you feedback? Is it And and like recognize it. And unfortunately it's one of those things like Murphy's law, like even you're going wrong when you even when

you take into account Murphy's law.

But if you if you don't name it and recognize it and discuss it and train yourself to think in that way that you know, take a breath, pretend you're a regular person, and then look at this again and see if it makes sense. I love You're screwed. I love that you threw this in here. I have a billion other questions I I'm going to ask you in part two when we do this someday. Uh Stewart, thank you so much for doing this. Thank you so much for being here.

Yeah, thank you for having me, Lenny. I really enjoyed it. Same same here. Bye, everyone. Thank you so much for listening. If you found this valuable, you can subscribe to the show on Apple Podcast, Spotify, or your favorite [music] 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](http://lennyspodcast.com). >> [music] >> See you in the next episode.