EXHIBIT 15

andreessen.horowitz

TABLE OF CONTENTS

A16Z PODCAST

Learning from Open Source Communities

Sonal Chokshi and Nadia Eghbal

SHARE ⌄

a16z Podcast | EP671

Learning from Open Source Communities

▶ 00:00

Posted August 11, 2022

We're living in an unprecedented era of online collaboration, coordination, and creation. All kinds of people are coming together — whether in an open source project or **company**, an R&D initiative, a department in a company, a club or special interest group, even a group of friends and family — around some shared interest or activity. But the word "members" is faceless, and doesn't help us really understand, support (and better design for) these communities.

So in this special **book launch** episode of the a16z Podcast, **Nadia Eghbal** — author of the **new book** *Working in Public: The Making and Maintenance of Open Source Software* published by **Stripe Press** — shares with a16z editor in chief Sonal Chokshi the latest research and insights

from years of studying the health of open source communities (for Ford Foundation), working in developer experience (at GitHub), researching the economics and production of software (at Protocol Labs), and now focusing on writer experience at **Substack**.

Eghbal offers a new taxonomy of communities — including newer phenomena such as "stadiums" of open source developers, other creators, and really, *influencers* — who are performing their work in massive spaces where the work is public (and not necessarily participatory). So what lessons of open source communities do and don't apply to the **passion economy and creator communities**? How does the evolution of online communities — really, social networks — shift the focus to reputation and **status as a service**? And what if working in public is also about sharing in private, given the "**dark forest theory of the internet"**, the growing desire for more "high-shared context" groups and spaces (including even podcasts and newsletters)? All this and more in this episode.

Show Notes

What "open source" means [1:56], types of communities [4:17], and how they control growth [7:19]

The modular nature of open source platforms [10:16] and the ideological framework driving open source software [12:48]

Further discussion of managing growth and the creator's time [15:12]

Open source contributors who create their own brands [20:10]

Discussion of platforms that are abandoned [22:34]

Subscription models and building an audience [26:51]

Platforms that are deliberately outside the mainstream [31:27] and their relation to newsletters, email lists, other semi-private spaces [34:43]

Crisis of the commons and how it relates to online platforms [36:37]

Guidelines for community managers, platforms, and communication tools [42:15]

Transcript

Hi, everyone. Welcome to the "a16z Podcast." I'm Sonal, and I'm super excited to do one of our special book launch episodes, for the new book coming out just this week — "Working in Public: The Making and Maintenance of Open Source Software" by Nadia Eghbal, and published by Stripe

Press. The topic actually applies to all kinds of communities and groups coming together, whether
it's an open source project, an R&D initiative of a department in a company, a club or a special

interest group — even a group of friends and family, because it's all about how people come
together to coordinate and collaborate around some shared interest or activity — whether
participatory or not, whether code or content.

And so, one theme we also pulled the threads on in this episode is about how the learnings of open
source communities do and don't apply to the passion economy and creator communities as well.
Nadia has long been immersed in studying the health of communities, including getting funding from
the Ford Foundation to study open source, then worked at GitHub in developer experience, then did
research at Protocol Labs, and is now focused on writer experience at Substack.

For longtime listeners of the "a16z Podcast," I've actually had her on the show years ago — along
with Mikeal Rogers of Protocol Labs, then of the Node.js Foundation — where we talked about the
changing culture of open source. You can find that episode on our site. But in this wide-ranging
hallway style episode, Nadia and I cover everything from types of communities, social networks, and
the evolution of being online. And, ironically, while the book is called "Working in Public," we also talk
about the emergence of private spaces, as well as the tragedy of big public commons — and how to
counter the tragedy of commons, which is why I believe everyone should read this book. Because
there's a dearth of literature out there for the era of unprecedented online collaboration, creation,
and consumption that we're in.

We end with some quick practical advice for community managers, platforms, and leaders, but we
begin by quickly defining open source in this context, with a really useful taxonomy for categorizing
communities.

Defining "open source"

Nadia: You know, early on, I was just like, "Oh, I really hate this term." And I just wish we could have
gone with something else, like public software or whatever.

Sonal: Ooh, I love that.

Nadia: I love it, too. Unfortunately, it's pretty hard to change terms that everyone else agrees on.

Sonal: Yes, I know this firsthand.

Nadia: Yeah, I mean, I personally find the term kind of intimidating. And I don't know, it doesn't sound exciting when I say the term open source. But it really does just refer to the distribution side of code. The existence of open source licenses made it very easy for anyone to use and modify and republish someone else's code, then put it in their own software. But it doesn't really say a whole lot about how open source is actually produced. And so I make this analogy in the book — which is actually an analogy I borrowed from my friend Devon.

Sonal: Devon Zuegel — she's hosted a couple of podcasts for me. I love her.

Nadia: Yes. And she says something like, "The term open source doesn't mean anything, any more than the term company does." It's like, yes, we kind of get what a company is. But there are so many different kinds of business models for different types of companies — and so, similar is open source. Saying something is open source tells you a little bit about how the code might be used, but doesn't really say anything about how they're actually being made. Someone has to continue taking care of it.

Sonal: One of my favorite parts of the book is how you actually outline different types of communities. You call it classifying project types, but it's really, to me, how people organize and, like — essentially social networks, really. So why don't you break down that taxonomy. And, by the way, the reason I'm asking is, because when I think of the arc and history of open source, the concept that comes to mind for everybody is that classic book by Eric Raymond, which is "The Cathedral and the Bazaar," and I think that framing has too long framed our discussion of open source, and frankly any online community.

Nadia: Yeah, everyone sort of has this maybe, like, general understanding of what community is — like, there's a bunch of, like, members, and they're kind of organized around some common interests, or reason for spending time together. In that highest level definition of a community, there's an underlying assumption that, like, all members are sort of similar. And just the term "members" sort of, like, washes over the underlying dynamics between those different members.

And so, what I started by doing was saying, "Okay, there is a difference between — at minimum, in open source — people who are contributing to open source, and people who are using open source." So I try to sort of separate out users and contributors and say, okay — in some open source projects — or, as you said, really communities in general — some communities have high contributor growth, and some communities have high user growth. And then there's sort of, like, different permutations of that.

Types of open source communities

Sonal: It's like federations, clubs, stadiums — and I forgot the fourth — toys. And tell me what those are. So I think it's really useful to start with your taxonomy of federations and onward.

Nadia: So federations are like the really big open source projects we might be thinking of — like Linux or Wikipedia — where you have a lot of people who are contributing to the project, and you have a lot of people that are using that project. But there are enough people that are working on the creation of that project that, like, it does form its own sort of contributor community. By contrast, clubs have a lot of people who are participating in its creation, but they don't have as many people that are using it. And so that product that's kind of focused on a niche interest — the example I like to give is Astro Pi, which is a Python library for astrophysicists.

Sonal: Right. It's high contributor, because they're incredibly interested in that, but very low user growth — because how many people in the world are really interested in that intersection?

Nadia: Exactly. And toys I sort of mentioned in passing, and they're probably the least interesting thing to talk about. That's where you have both low user and low contributor growth. So that might just be, like, a personal project that I'm tinkering around, no one else is really looking at it. They're sort of waiting in the wings before they become one of the other types of communities.

And then the fourth model is stadiums. And this is the one I think is most interesting and most overlooked, because it's kind of a newer phenomenon. And so this is a situation where you have one or maybe a couple of contributors, and then you're making something for, like, a very large audience of users. And so you can imagine someone's standing in the middle of a stadium — there's this imbalance where, in this case, the developer is feeling a lot of inbound requests, a lot of comments, issues, pull requests — just a lot of needs from their users.

But there aren't that many people who are actually able to help. Contrast it to a federation, where you imagine something like Linux — is extremely widely used, but there's also a very mature and well developed ecosystem of contributors to support it. But a big part of this book is taking the time to stop and look a little bit more deeply at — what is that giant audience of faceless users, and are there interesting dynamics happening there that actually make this look more like a community. Where, like, a stadium is actually a legitimate type of community that stands alongside the clubs and the federations. We just haven't really taken the time to understand it before.

Sonal: I love that, Nadia. What I found fascinating about stadiums is — you're essentially describing — and I think about this as someone who cares about content and social, is a rise of an influencer. This is no different than influencer economies, in many ways, where you have sort of, like, a star

contributor and then, like, a bunch of people kind of in this stadium — literally, in your analogy, watching them. And you even say that it's this shift — and I think you're quoting someone — to facing the stage, versus facing each other. So when you have this person who's on the stage, and they're, like, the primary contributor — and let's just say creator, because we're essentially also talking about creator economies here. <Yes.> You made this distinction that they may be intertwined and influenced by their community around them, but they're not actually "doing peer production" in the classic collaborative way of the first eras of open source. So can you explain that a little bit, and tell me a little bit more about why that's happening?

Nadia: Definitely, it does have this broader application to what's happening to individual creators on all these different social platforms today. Most open source projects, we can probably say, used to be clubs — where just, like, not a lot of people were using open source in its earliest days. You kind of had this group of weird developers who just loved using it and maintaining it for their own purpose. And then, eventually, we kind of hit this point where open source became so popular that tons of people were, kind of, discovering all these projects and using the code. And I would attribute that in large part to the creation of a platform — GitHub — which kind of united all these projects together and made them discoverable in one place, with a much more standard way of contributing and discovering — and just, like, thinking about what is open source.

For a lot of people, GitHub is basically synonymous with open source. Another useful parallel trend here is just that — open source projects started becoming a lot smaller, due to just platform effects of different languages having these package managers that made it really easy to find and discover and use lots of different libraries. And so, now these projects are smaller. They have one developer at the helm, but they have 10x, 100x, 1,000x more users that are coming in.

And so, suddenly you go from having these clubs where everyone kind of, like, knows each other — if you're using it, you're kind of expected that you will be contributing back if you need something, instead of asking someone to do it for you. Suddenly you have, like, all these outsiders that are, like, flooding into a project and using it. At the same time, you also have some portion of those users who are now coming into the project and asking for things. And they don't have the same background that the core developers or creators or maintainers do. They're just sort of, like, asking for things and leaving.

A useful analogy here might be thinking about a small town that was largely undiscovered, was not connected to a highway — and then once it became part of a highway system, then you suddenly have all these tourists who are now stopping by some cute little town. And suddenly, it changes the nature of the entire town. Because, I mean, in some tourist towns, you can have — more than half the population is actually tourists, and not even local residents. It's either, "I'm just gonna completely

close off and do my own thing, or have to, like, welcome everyone." And those are kind of, like, the two extremes that I often hear about when they're trying to, like, think about, "How do I manage this volume?"

And so, what I'm sort of trying to suggest is there's something in the middle there — where it's okay to make things and share them in public. But it doesn't mean that everyone has to participate. And that's a theme that I really tried to push on in this book — is that something being public does not mean that it has to be participatory.

Sonal: You actually shared a great analogy in your book, where it's like the Twitter user — like, an early Twitter user, before they become kind of famous or big. They're very good about, like, responding to people. They're building their community. It's very, like, peer-to-peer. And then there's a point where some of them become even more influential. And they're so overwhelmed by mentions and replies and questions that they can't even remotely respond to any of it — let alone little of it. So I thought that was a very useful analogy for thinking about that. Because one can also evolve over time.

Modularity of open source platforms

So you mentioned that there's this, kind of, increasing packages — where people can kind of take things and combine them. And this really stood out to me, because one of my absolute favorite themes when I think of, sort of, meta themes for innovation, and how people change the world, and how people change things — is modularity. And I have this, kind of, joke of, like, "Modularize all the things." Let's talk about why that modularity matters. And the example that we both know is modularity in the form of, like, the node package manager. Our mutual friend, Mikeal Rogers, ran much of the work in the Node.js community. Let's talk about how that shift has mattered.

Nadia: So, on the one end you have very monolithic software, where if you change one thing, it has a lot of patience for changing other things. Software products that look like this tend to be a lot more thoughtful and deliberate and slow about what they actually want to accept as a contribution, or changes that they want to make. Because the whole thing is tightly coupled — but also, sort of, brittle in that way. And so it has just a very different implication for, like, how many contributions we actually accept — how much can we actually change things?

What happened when open source became a little bit more modularized — which is probably best exemplified, as he said, in the story of npm and JavaScript — is that now instead of having this tightly coupled code — you can imagine, like, a tower made out of Lego blocks. Where you can remove one of those blocks, and, like, the rest of the tower still stands. So, it completely just, sort of, changes how we think about a single piece of software. And that, like, instead of having to think

about the major implications of changes between the different parts of the code, you can actually say "Hey, I'm gonna grab, like, lots of different components from different types of developers. I want this person's library and this other person's library" — and just, like, fine-tune it to look exactly how you want. And as a result, it enabled a lot of new creation in open source.

Sonal: In the crypto world, the community and the team here loves to talk about the composability of open source projects. Composability being the idea that you can take these building blocks — you mentioned Legos — and that's really important, because people are combining, remixing, and reusing. And it's kind of a buzzword, but I use it on the podcast — I'm gonna stop being ashamed of it, which is — combinatorial innovation. And it's very "primordial soup." Like, you get all these ingredients, and then it leads to this combining and recombining and evolution, and the Cambrian explosion. I'm just throwing [out] a ton of buzzwords there. So, that's why the modularity matters. So now, can we talk for a minute about what it means from a project point of view? When open source goes from big to this small kind of — collectives of people that may come together, what are the implications of that?

Nadia: So, if you talk to free software activists from, let's say, the '80s or the late '90s…

Sonal: I used to edit one of them — Richard Stallman. And he would call that "Libre."

Nadia: Yeah, people that are really focused on the sort of, like, ideological implications of open source or free software — if you talk to them, you'll find that — or I, at least, found — that a lot of them are really concerned about the liberation and protection and longevity of the code itself. Like, freedom is not referring to any freedom of developers. It's referring to freedom of the code. <Right.> But if you kind of come down to, like, a world where things are a lot more modularized, suddenly the focus shifts from the code to the people who are behind it — because now every piece of code is much smaller and more trivial.

There are very well-known developers, especially in the world of JavaScript, where that really encourages a lot of this, sort of, style of development. There are very well-known developers who make hundreds or thousands of popular npm modules — which are each their own separate project, but they're — each one's very small. And so, suddenly it kind of becomes more about the person behind it. A useful parallel here might be thinking about the impact of tweets versus blogs, where a blog post is this, like, lengthier thing, and a blog post kind of stands alone as this beautiful piece of literature or whatever.

But then, like, if you're really into using Twitter, like, you might tweet like 100 things in a day — and one tweet might go viral, but like, you have so many more that come up, right? And so, it kind of just becomes, like, about the person tweeting. It's not about, like, "Oh, he wrote that amazing tweet six years ago that I often revisit." Like, that's not really what it's all about. And so, I think — to sort of summarize this — I think this shift towards modularization also helps drive why we're seeing more interest in reputation-based and status-based economies. <Yes.> Because it just, like, wasn't the factor before. It was all about the code. Now, it's all about the people.

Sonal: Mikeal Rogers and I actually wrote a piece about this when I was at Wired. It was called "The GitHub Revolution." And this was like in early 2013. And, basically, the fundamental point is that GitHub inverted the model from project to person, and then identity came [into] the picture. But to your point, when you have these modularized packages, and individuals who are very tied to that, it does become about the person. But now, on the social side, if reputation and the person is at the center — not just the code — what does that mean for how these groups organize? And what does it mean for how they manage and how they collaborate?

Managing growth and the creator's time

Nadia: Yes, there are absolutely different implications for how these different types of communities can and should think about organizing, and how they think about growing and maintaining over time. The currency that I've settled on was focusing on a producer's attention as a limited resource. So we all talk about the attention economy, but the attention economy tends to refer to a consumer's limited attention. But we don't often talk about a producer's limited attention. So, like, a creator only has a finite amount of time as well. If we're thinking about creators and not these, like, big distributed communities now, the creator is kind of, like, on their own, and their attention is not gonna scale by themselves.

The first line that I would draw is between clubs and federations, which are dealing with an abundance of attention, because they can be high contributor growth. And then stadiums that are dealing with dearth of attention, I guess you could say, because their contributor size is not growing significantly, but their number of users is growing.

Sonal: Right. And just again to emphasize, you're talking about the attention of the contributor and the creator?

Nadia: Yes. And the ones that are probably most interesting to talk about today are the difference between clubs — which have high contributor growth and low user growth — and stadiums, which have high user growth and low contributor growth. And so, one of the things that previous online community literature focuses a lot on — and especially also in open source — are governance

processes. And governance is probably more useful and important to talk about in the context of larger contributor communities, because these are coordination problems, right? Like, you have multiple members with a stake in the community who are all coming in with their own interests. And you're looking to figure out, like, how do we all best work together?

On the creator's side, there's probably another version of these processes that need to be developed for stadiums that's not really about governance, in the same way, because you usually only have one or a couple people that are at the helm. It's more about the relationship between that creator and their audience and, like, "How do I interface with my audience? How do I make them feel heard? How do I utilize people that might be willing to help or pitch in?" So there's a lot of just, like, different kinds of strategies they can think about around, like, how do I — given my limited amount of tension, like, how do we make sure that stuff continues to get done.

Sonal: Right. To pull on a couple of other threads there — does this mean that these relationships even have to be persistent? I want to hear your thoughts on that, because we talk about these very stable federations that have been around for decades. But one thing that I find very appealing — and might be a bug to you, but I think is a feature — is that some of these things seem like they don't have to be persistent and can maybe be very ephemeral, when you have that kind of small modular setup.

Nadia: I absolutely think the relationship between creators and their audience becomes a lot more ephemeral. And we should almost be, like, leading into that design, right?

Sonal: Yes. I really strongly believe this.

Nadia: Yeah. And so, like, there have been these terms that have existed in open source for a while — the idea of, say, like, a casual contributor — to distinguish between someone who's kind of dropping in and making one contribution, versus someone who's a more, like, active or present community writer.

Sonal: Right. Didn't we even call them — I think in our last episode — they're drive-by contributors, right? <laughter>

Nadia: Yes. Drive-by contributors, casual contributors. And so these are the people that are not coming in with a pro-social attitude. One thing I did find in my research is that folks that come in as these more active contributors making substantial contributions — a lot of them do come in displaying pro-social attitudes from the beginning.

Sonal: Ah, interesting.

Nadia: Yeah. So they are coming in saying — they're looking for a community that they want to be a part of, and they want to help out. So, like, one behavior you might see that's different about an active contributor versus a casual one is someone coming in and, like, answering someone else's question, instead of opening an issue saying, "Fix my thing." It's, like, two very different kinds of behavior, right? Like, one, you're trying to help someone else — and, one, you're asking for help. Like, "I want to get something out of this. I want to get my contribution merged. I have a question that I need answered," whatever. They're coming in with some sort of personal interest.

Sonal: By the way, you also use the word parasocial in your book, which I had to look up because I didn't even know that was a thing.

Nadia: I actually think, like, parasocial is a great way to just describe what kinds of community these stadiums essentially are — which, it basically just means, like, one-sided communities. Where, like, one side of the audience has a deeper, more perceived intimate relationship with the creator than the creator does to them.

Sonal: That's very similar to podcasting.

Nadia: It is very similar. If a creator were to treat every single fan that they met or every single person in their audience as someone that they're gonna develop a deep and long-lasting relationship — like, that's just exhausting. It's completely impossible. But if they say, "Okay, like, we are gonna just meet this one time. Like, how can I make sure that this person feels fulfilled, or whatever, and I manage this without giving too much of myself?" And so, yeah, like, these interactions are more ephemeral. And we can, sort of, design around that where, like — "Here are a bunch of, like, self-serve resources." Or we can encourage users to help each other, instead of always turning to the creator for help. And so all these other, sort of, like, supporting satellite communities can thrive and flourish on their own without needing an involvement from the creator.

Open source creators and branding

Sonal: What do you think of something like "The Ringer," where you have someone like Bill Simmons — the analogy here is, he's a hotshot coder — but, really, he's like, a creator. He did "Grantland," and then he went out on his own and did "The Ringer." And then within that, he built a constellation of brands underneath his parent brand. It's both bundling, and also, like — just constellation communities. Do you have thoughts on how that works? And how that might play out in the open source world as well?

Nadia: Well, I guess there is a version of that that happens in open source, which is — you have this broader language ecosystem. I'll keep coming back to JavaScript as the best extreme to demonstrate this. And we can drill even further into JavaScript — let's say like the React ecosystem. And within React, there are a bunch of associated projects that a React developer might use. And so when we think about who is a contributor to that project, like — yes, you could look at who has actually made contributions to some specific subproject. But you could also say, "Well, who's contributing to, like, React more generally?"

And so taking, like, webpack — or something that is a subproject that a React developer might expect to use — someone might have never contributed to Webpack before. But if they're well known as a developer in some other part of the React ecosystem, then they already have a little bit of currency and a little bit of reputation if they were to try to come in and open a pull request, or make a contribution. And so I don't know exactly what the analogies are between that and sort of, like, subscription bundling, or what that can look like.

But one thing might just be that when we think about — what would it look like to have more subscription-type support for open source developers — which GitHub Sponsors, Open Collective, there are examples of this already — we might think a little bit more about, well — it's not just this one project that this developer works on, but they work on this ecosystem more generally. And so, maybe similarly — the way that, like, a writer might have started with, like, one type of newsletter, and then, like, they join forces with another one — and then, suddenly, we're sort of supporting this entire bundle of people that are working on a similar theme. You can imagine that happening with open source developers, where they're no longer just tied to, like, one specific project, but it's like, "I support your development work more generally."

One of the more obvious examples, I guess, I could point to is Sindre Sorhus, who has done pretty — like, thousands of mostly npm-related projects. <Right.> But he's sort of, like, his own mysterious entity. It's not really about any one specific thing that he does. <Right.> He's just, like, a very generative person. And he is supported through sponsorships.

Sonal: I'm gonna ask you a crazy question. This is a thing I've been very fascinated by for a long time. I tried pitching this at Wired — by this idea of, like, digital suicide. Taylor Lorenz writes these beautiful pieces about, like, Instagram, and all these various communities online, etc. And I'm also fascinated by this phenomenon of all these, like, teens creating multiple accounts and multiple identities on their Instagrams. And then they also abandon them, which is something I love — this idea of this kind of abandoned wasteland of digital identities and places, because it feels like the real world to me. That there are places that are ghost towns, and places that have been lost in the sands

of time, for better and worse. Do you have thoughts on how that may or may not apply to open source, because not only do these things not have to be persistent — they can be ephemeral. Is it okay that they die, or that they even have — up front — a calculated, kind of, end point?

Nadia: Oh, this is where software gets really interesting — and, I think, different from most other forms of creation. Because if someone creates an Instagram account that gets really, really popular, and everyone's following it, and then eventually — suddenly — this person goes dark, and we never get another post of them — a lot of people will be sad about it. People might create, like, spin-off accounts, and tribute to that original account, whatever. But, like, the world doesn't actually, like, break and shut down.

If a maintainer has a product that is wildly popular, and they're just sort of, like, over it, and they disappear — and this does happen often — that code is still — if it's popular, is being used by a bunch of other people. And, like, code changes over time. It doesn't need upkeep and maintenance. Intrinsic motivation helps a lot with — on the creation side of things in the very beginning. If something becomes really popular, then you start getting these more extrinsic rewards, like reputational benefits, or status, or whatever. But a lot of stuff is sort of front loaded.

And so, if you're talking about maintaining a software project into perpetuity, after a while, you're already known as the creator of that thing — there's not really any additional benefit. But you still need to keep maintaining, and, in fact, sometimes those maintenance costs get even higher over time. And so, that's why I think it helps make the case for — we need to find other reasons to keep people wanting to maintain stuff or make it easy for them to step down. Because intrinsic motivation really only takes you so far. And so, if something happens, where they actually, like, need some changes to be made or need updates to be made to the project, and the maintainer is nowhere in sight — and nobody else has the ability to, like, make commits or contribute to the project, like — this actually can create, like, real problems for software.

Sonal: You're basically describing software as more of a living, breathing organism, actually, in that context. <Yes.> You either want to evolve it and keep it going, and generationally it can evolve into something else and have offspring, etc. But it's a different thing than when you just have, like, an abandoned site or, like, an abandoned farm somewhere.

Nadia: And this is why, like, from the beginning, the ability to fork code or basically, like, copy the exact repository somewhere else — has become — it was, like, a very important part of it early on to say, like — someone can always take the copy of this code and make their own version of it somewhere else. Unfortunately, this comes back to this dependency issue today, where — yes, in

theory, you can fork a project. And in practice, there might be a lot of other things — other software libraries, other pieces of software — that are pointing to that specific project. And so, if you fork it, you now have to somehow convince all those projects to start pointing to your new project.

And so, it is this challenge with open source where sometimes, like, a maintainer disappears and is nowhere to be found. But, like, they still need to keep doing things to the code and to the project. Forking is not always an answer — an easy answer to that. It's still about telling everyone like, "Hey, come over here. Use this." And that's actually why I think this concept of maintenance, that is maybe easier to see in software, actually really applies to every creator today. <Yes!> Because a lot of people go through this experience of, you know — you did one thing that might have gotten you this huge, like, seed initial audience or whatever.

Sonal: It made you internet famous too, for lack of a better phrase. <crosstalk>

Nadia: You can have that moment, but then you know, you have to continue creating things — otherwise, people are gonna stop paying attention to you. Much like writing code, it's not enough to just sort of, like, publish it once and be like, "I'm done. Like, I'm never gonna touch this thing again." If you are trying to build this reputation over time — and some people will say, "I had one viral video on TikTok, and that's it. Like, I'm never doing anything else again." But if you're trying to be, like, a TikTok creator, you're gonna have to start making more hits over time. And so your reputation is itself this thing that requires maintenance in order to stay relevant.

Building an audience

Sonal: It is literally one of my favorite parts of your book, because it reminds me of the theses that we've talked about at our firm, too, around the passion economy. Here, the artifact is code, but it can be any activity that's being coordinated, quite frankly, in your framework of your book — which is why I really believe, again, that this book is applicable to everybody. Open source is almost a misnomer, because people think it means code. And it really means everything. It's like any kind of creation and consumption, frankly — but what you're really saying is a maintainer is not just a coder, it's a creator. And they're maintaining their content in this world, or whatever they're creating — which I think is incredibly powerful.

And what's really powerful about that is — then you think about sort of the related business models for that. Like, when I think of the example of what subscription and SaaS, Software as a Service, did for the world of on-prem software and how people used to sell software — and you had, like, the suited person do this big multimillion dollar deal install, never see them again. SaaS changed the game for everybody in companies because you had to consistently earn their dollars every month, but in a way that was a wonderfully sticky, stable relationship too, like you were mutually dependent.

Nadia: Yes, this is, I think, a giant red arrow pointing at why subscription models are gonna only become more and more interesting in the very near future, because they do take into account this need for, like, ongoing development. There are ongoing costs associated, and you have to earn that, as you said, over time. And so, they're capable of sort of capturing both your existing value — the value that you have accrued thus far — and also speaking towards the future value that you might create, because when you subscribe to someone or someone's thing, you're saying, "Like, I expect there is gonna be more stuff being created in the future."

Sonal: And this goes back to the phrase, and something you said in your book, and that's sort of theme for me — you can be transactional, but be in a very high sustained relationship, because it's a repeated game — which is what subscription is. I think that's super fascinating. So what do you think the implications are, then, for people who change clubs. They go to a different — they create a new stadium. Like, how has this new passion economy and model evolved?

Nadia: I think this kind of comes back to the value of platforms. And they're a distribution power. So, I mean, in theory, in the past — without having a platform, which is essentially just a stage for creators. That is always gonna exist, is always there for the creator — without that, if you wanted to go off and, like, start something new somewhere else, it's really, really hard. Because how are you gonna direct anyone to your new thing all the way over there? <Right.> It's like building a house but not building a road to the house.

Whereas, like, platforms have this very important role that they play for creators, where if you want to do something new, you have an audience that you're building on there that you can use to seed whatever your new ideas. Platforms make it so much easier than you could have in the past. I mean, this is also — so I work at Substack, and this is also why I and everyone that I work with believe really strongly in the power of an email list, because an email list is something that you own. And if you want to do something new with it, if you want to do something totally different, like — you have an audience that is sort of built in and that you can take around with you wherever you want. But even if you don't have an email list, like, having a Twitter following or having an Instagram following, or whatever, gives you that sort of, like, seed money to do something else.

Sonal: Basically, you're saying that you have the distribution because your audience travels with you. And that's an important currency because you don't have to start from scratch every time. That does go to your other point as well, that the reputation is the key and the currency there. And that's where status — and you talk about this in your book, and Eugene Wei's thesis about "Status as a Service" comes in.

Nadia: Eugene's thesis came out, thankfully, while I was writing my book, and it was very helpful, because I was like, "Okay, now I have more vocabulary to explain the things that I mean, that I've been struggling with." I think actually this framing of status economies helps explain some of the shortcomings of GitHub thus far, because there isn't, sort of, a meaningful way to measure someone's status — or just have a clear picture of what someone does on the platform, or what kind of developer they are. You can look at any one specific project, and you can see how popular it is, you can see how many stars that it has. But if you go to a developer's profile, it's not super clear what they're known for. You can technically follow a developer on GitHub, but it doesn't really mean anything — not at all the way that it does on Twitter or something like that.

And so, I think if you talk to well-known developers, or developers that have these larger followings, they'll probably tell you that they keep their audience on Twitter or somewhere else. And GitHub serves a little bit more of this utility function, as Eugene said. Where if a platform fails to provide this sort of status benefit, then it basically becomes a utility. They will continue to develop the social and status aspects of their platform. But right now it really is much more of a utility, I think.

Sonal: So, you mentioned the power of a platform. And you've been using this analogy of, like, cities, and highways, and connecting houses, and connecting a small shop or a small village to a highway — and what that does for people. What about the opposite, when people go off the grid, essentially, and go outside our purview into these sort of private, dark social places — whether it's WhatsApp groups or Telegram groups, or private stadiums, private groups. And you mentioned in the book — and I saw Yancey Strickler's tweet about this when he did it on Twitter — he's a former co-founder, CEO of Kickstarter. He draws the analogy of the dark forest. The reason that we can't communicate with aliens is because the world is so vast, and the only way people can protect themselves is by being in this dark forest, where there are these vast spaces of separation. So you're not in this vast — you're not actually in what is commonly referred to as a public commons. You're actually very isolated.

Nadia: The only thing I would maybe add there to the dark forest concept or metaphor, is this idea of hostility — that we are all, actually, surrounded. There are all these other people out there. If we're sitting here wondering, "Where are all the aliens?" They're there, but the theory being that we're all trying to stay out of each other's way and not be detected because…

Sonal: Destruction will be the result.

Nadia: Yes, it's not a good thing to meet anyone else as curious as we are.

Sonal: The dark forest comes from the idea of the Fermi paradox, I believe — and I'm a big fan of "The Three-Body Problem" trilogy. They have the wall facer — he's the one who figures this out. So I thought that was a super interesting analogy. Tell me a little bit more about your thoughts about the dark force theory of the internet, and how that applies here. What happens when people go off platform?

Nadia: So Yancey Strickler's comment about this. And I think basically a lot of people are observing that, okay — we started with these really big social platforms that have grown to become really big — so, the Facebooks and the Twitters and the Instagrams and YouTubes of the world. These are sort of, like, the biggest stages possible. And so, the analogy to what's happening on the very public web right now is that everyone's still talking, it's just sort of, like, we're kind of moving to these little corners, without fear of being attacked or jumped on somewhere where all context has otherwise collapsed.

Sonal: I'll say one more thing, because I'm a big fan of the work of the sociologist Ronald Burt — and he talked about this concept of structural holes, where you can have, like, clusters of activity and networks. And I came across this because when I used to work at Xerox PARC, we used to talk a lot about the innovation that would happen when different fields would collide. And it's because you have these containers — these clubs, these stadiums — of people who have strong ties, but then these really interesting things can happen with what are the weak ties, and then the structural holes in the network. So, if you map these out as, like, a universe of clusters, imagine what's possible when you can actually bridge some of those structural holes across communities. <Mmm.> Like, your book made me think about that, actually. I wonder if that's where the future is going. Is a bundle maybe that? Who knows what's happening there? I mean, we're only at the beginning of it.

Nadia: Yeah, I mentioned this quote in the book, but Kevin Systrom said in an interview in 2018, I believe, that social media is in this pre-Newtonian age where we know that it works, but we don't know how it works.

Sonal: Ah, I love that.

Nadia: I just think that's really perfect.

Sonal: It is. It's perfect for the time we're in. And it's perfect for why your book is so relevant.

Nadia: This is where I think the model of clubs versus stadiums becomes really useful. For a long time, everyone was really focused on, like, the highly public aspects of the social web. But people are now starting to look at the semi-private web and these quieter spaces. The biggest parallel

trends that I'm seeing right now — like, one is seeing this formation of these creator-oriented communities that look like stadiums on the big public stage — in, say, like Twitter or whatever. But then you see this other emergence of, like, group chats. And group chats have become this really — I mean, [they] have always existed and [have] kind of become a much bigger thing in recent years, <Yeah.> partly because people are looking for a relief from this high, heavy public space. And those map really well to clubs, where you aren't trying to add a lot of users to your messenger app. You're trying to just keep it to, like, six of your closest friends. In most cases, we'd say that you're, like, actively suppressing user growth. But contributor growth is high, where you're totally down to chat with your friends and that little group.

So, those map really well to the clubs that I sort of identified here. Whereas stadiums apply to both these, like — creator communities are happening in very public platforms but I think can also help us understand why things like podcasts and newsletters are having such a great moment in the sun right now. Because they're designed for that one-sided, parasocial type of community. <Mmhmm.>

Where if, you know, we're recording a podcast right now, it's just a conversation between me and you. And, yes, hopefully, thousands of people will be listening to it later. But that we're, sort of, like, doing this in public — meaning that we're publishing our conversation — but we're not actively interacting with the audience that might be listening to us. And similarly, with a newsletter, I can write this long-form post and share my thoughts in a higher-context situation. I assume or hope that most of the people subscribing have some context for who I am. And then I can, kind of, send it out, and people can read it on their own time. It's not the same thing as when I tweet something out, and then literally anybody with an internet connection… <Right.> I made a public tweet — can see it and respond to it and pass it around and do whatever they want with it.

Crisis of the commons

Sonal: And so, to summarize, the clubs are the projects — the spaces with high contributor growth and low user growth, like these private messaging groups. The stadiums are, like, the projects with a low contributor growth and high user growth, like these newsletters and podcasts. I really think, Nadia, one of the best things about your book is this framework of the federations, the club, the stadiums, the toys — because you dehomogenize this phrase "open source and community." And then it, correspondingly, gives people frameworks for what that means for how you build, support, nurture that.

So, I'm now gonna switch to asking you some practical questions about that. Platforms are having a moment right now, for better or worse. It's one of the reasons that we also are very excited about crypto and talking about communities. And I want to talk about the tragedy of commons and the work of Elinor Ostrom, who is definitely having a moment. Right now you and a lot of other people I

know have been citing her work. One of our former partners, Jesse Walden, wrote a post about cooperatives as an analogy for crypto networks. And he cited some of the conditions that she cites in governing the commons — and then you yourself summarize the conditions. Can you, A, tell me what those are — B, tell me why you think this is important, and then help me connect the dots for how that matters practically?

Nadia: Sure. So, Elinor Ostrom was a researcher who became well known for her work around trying to understand why the tragedy of the commons occurs, and how we might avoid it or move around it. Tragedy of the commons just sort of being that — if everyone has access to a shared resource. You can imagine a fishery or a forest — anyone can cut down wood in the forest. But if everyone does that, and just, kind of, does what they want for themselves, then eventually that forest is gonna be depleted unless it is managed in some shape or form.

And so, tragedy of the commons is this concept from ages ago that is, maybe, one possible outcome of the commons. But it's almost like when people talk about the commons, they always talk about tragedy of the commons — as though you can't have, like, a non-tragedy of the commons. And so, Elinor Ostrom is basically looking at — what are situations where commons are being sustainably self-managed. And she did decades of research looking at these, like, fisheries and forests, and just, like, different examples of commons, and then documenting what she found and summarizing them into principles for — if you are in the situation where you have this shared resource, how can you manage it without everyone just sort of taking for themselves.

And so, I talked about her conditions in the book a bit, and the ones that I'll point to that are most relevant for this conversation are — this idea that in order to have a well-managed commons, you do need to draw boundaries around membership. It needs to be clear who is allowed to appropriate from the commons and who isn't. And then with that, there are all these implications of, well, what does it mean to be a member of the commons? A couple of things that I'll highlight are, one, this idea that you have high context for your interactions with other people that are also members.

Sonal: Yep, that creates trust. That's what creates trust. It's just like in a company. They say the best advice you can give to any team or any fast-growing group is the more shared context, the more trust you have — because you can do more shortcuts together in your work.

Nadia: That's right. High context, high trust is a really important implication of having these clear membership boundaries. And then the other thing I'll point to is the idea of having a low discount rate — which is just saying that if you're a member of this community, you expect to be around for a while.

Sonal: Sorry, what do you mean by low discount rate?

Nadia: Low discount rate is just this idea that if you're invested in the community for a long period of time, you're not planning on hopping in, saying something rude — if this applies to online communities — and then just, like, hopping out and disappearing. You're like, "I'm stuck here. I need to, like, actually learn how to work with everybody else."

Sonal: Right. It's actually, kind of, like, skin in the game.

Nadia: Yes. In order for a commons to function in this healthy way, you need to have these underlying conditions of high context, of having skin in the game, of having clearly defined membership, among a bunch of other things that I won't get into here. Her work is finding, I think, renewed appeal right now — especially because people are trying to answer these questions in open source and in online communities elsewhere, of just, like — how can communities self-manage and not implode over time?

There's so much that is relevant about her work to today. I think it mostly applies, though, to the concept of clubs — clubs basically being this commons, where everyone has a stake in what they're creating. If we think about a stadium — a creators' community that is on a very public social platform — the whole concept of the commons kind of breaks down, right? Like, I mean, if I'm tweeting in public, anybody can read what I'm saying. And until recently, as Twitter's now making it possible for people to limit comments on their tweets, and things like that — but for the most part, like, anybody can just, like, comment on my tweet and jump in.

And so, understanding, I think, both what her theory of the commons was — and why it doesn't really apply to today — can help answer some of these questions about — is it okay to have common threads that are entirely open to everyone? What are the problems that might arise from that? And then what can we do to actually limit interactions from outsiders, so that the people that are most involved or have most skin in the game can actually get stuff done?

And it's a difficult thing to talk about, because it can be taken as gatekeeping or trying to keep other people from participating. It's just, like, a touchy subject. I can't say that we should just close off the boundaries entirely. And I think this gets to, again, the idea of — you can have things that are public but not participatory. It's okay to make software that anyone can use. That doesn't mean that everybody who uses your software can also participate in its production. So, it's really just about finding kind of, like, a middle ground there.

Sonal: You have so many great analogies in your book — that sometimes this is more like directing air traffic, given the flood of abundance we have on our internet today. So, on that front, I'm gonna

ask you just a couple of quick questions on the practical front. Let me do this lightning round style. What is one key piece of advice you might have for community managers?

Advice for platform management

Nadia: For community managers — first thought is, just know what kind of community you're in charge of. Which is where I think it's helpful to have a set of different models in your mind of — are you actively trying to bring in lots more contributors? It's okay if you're not. Some communities do better on contributor retention, and less so on contributor growth. And that's totally fine. Or is it the kind of thing where there is a lot of work that needs to be done? And do you think you stand a chance of recruiting more people? Then go recruit more people.

It's fine to have a community that isn't super high growth but is stable. It's fine to have a community that is extremely high growth, where you're trying to bring in lots of different kinds of members and make it this really bustling, kind of, federation-style community. It's fine to have just one person that is, sort of, standing up in front of a crowd. That is a community in its own form, but it just requires different sorts of strategies to figure out how to manage it.

Sonal: You're basically saying — know the difference of whether you have a club or a stadium. And, by the way, you quoted that person talking about the Newtonian phase. Who knows? There might be times when you can have both in one place, so that can change. And then how about advice for platforms?

Nadia: For platforms, I would say — take your creators seriously and the responsibility and the relationship that you have to them. And what I mean by that is that platforms are really the only place to create these closed status economies that enable creators to continue doing their work for however long they want, and to open up all these amazing opportunities for creators. And sometimes that doesn't directly happen on your platform — as in, like, maybe it's not that they can raise money directly on your platform, but it is important to make their status legible to others so that they can take that clout and that reputation and actually, like, shop it around to turn it into other opportunities.

Sonal: I might even add that crypto is great for that, because that's where you can actually port some of your currency — your reputation currency — and prominence in a way that — like, in blockchains. And then, finally, advice for leadership and/or communication tools. Because we've talked a lot about — I think a lot of times people make the mistake of talking a lot about collaboration

and coordination. But they don't often talk about the communication part of things. And this is particularly heightened in our remote world. So, any advice you have on the communication tools side and then anything for the leadership side?

Nadia: More so the leadership side. On the leadership side, I think it's, again, about knowing your community and not being afraid to be decisive. A lot of communities that I've looked at have suffered in one direction or the other of — either being overly deferential to their community and trying to treat it like this pure democracy, when really the community's size or shape is just so unwieldy that that's not really possible. And so it is okay to say, "These are the decisions that are being made. And we don't have to make this — bring this to a vote every time we want to decide what we want to do."

And maybe also on the flip side, that depending what type of community you are overseeing, there are ways to bring those active voices and contributors into leadership and encourage more people to participate. But, again, it depends on whether you're on this, like, high growth side or low growth side. On the communication tools side, I think this idea that separating the ideas of public and participatory is just gonna lead to a lot of really interesting things happening in the near future. Just getting playful with the idea that a community does not mean that everyone is participating at equal volume and, you know, shouting at each other — because we've seen with, like, every social platform, that gets pretty hard at scale.

And so, like, as we're creating new things today, it's fun to think about the opportunity we have in front of us to actually design from scratch about — in thinking, like, how would we have these sort of scaled social interactions? And so disambiguating the idea of public and participatory can just lead to really fruitful new ways of communicating.

Sonal: This reminds me — one of my favorite quotes from Questlove. This is from his book, "Creative Quest." He basically writes that when you make work, you are the creator, but also the eventual audience — which I think is such a powerful idea. There's, like, so many different ways to interpret that. What I love about your book is that it's not a grand theory of everything. It ties together lots of different themes together in a really meaningful way. But you can also tease them apart regardless of your vantage point, whether you're a creator, open source, business — however, you might define a company or any form of coordination and collaboration.

I also appreciate, given that you were kind of newer to the community — compared to, sort of, the first, early generations — that you don't bring this sort of chip of nostalgia, and, sort of, come at it from a very first principles approach, and just sort of really bring all your insights together. So I just want to thank you for "Working in Public: The Making and Maintenance of Open Source Software."

I'm gonna add my own personal subhead, which is — and many, many other kinds of orgs. So don't just not read it if you don't think it's about you, because open source is everyone. Thank you for joining the "a16z Podcast."

Nadia: Thank you. It was a pleasure to be here.

Want more a16z?

Sign up to get the best of a16z content, news, and investments.

First Name *

Business Email Address *   SUBSCRIBE

MANAGE MY SUBSCRIPTIONS

By clicking the Subscribe button, you agree to the **Privacy Policy**.

Sonal Chokshi is Editor in Chief of Crypto at Andreessen Horowitz.

FOLLOW
X
Linkedin

Nadia Eghbal

MORE FROM THESE CONTRIBUTORS

Where Innovation Happens
Matt Clifford and Sonal Chokshi

What We're Reading, How We're Reading, Why We Read

Sonal Chokshi, Robert Hackett, Tim Sullivan, and Stephanie Zinn

TABLE OF CONTENTS

Conduct

Privacy Policy

Disclosures

X   in   f   ◉   ▶

© 2024 Andreessen Horowitz