EPISODE 1361

[INTRODUCTION]

[00:00:00] KP: Venture capital investment has continued to flow into technology startups. No one builds technology from scratch. There are cloud services, software libraries, third-party services and software platforms that modern entrepreneurs must adopt to build their products efficiently and quickly. These layers of infrastructure are a key area for many investors. In this episode**,** I interview Tim Chen, managing partner of Essence VC, a venture fund on a mission to help highly technical founders go from zero to one. We discuss his approach to investment in infrastructure companies, developer tools and similar areas for early stage investments.

[INTERVIEW]

[00:00:43] KP: Well, Tim, welcome to Software Engineering Daily.

[00:00:46] TC: Yeah, it's good to be here.

[00:00:48] KP: Tell me a little bit about Essence.

[00:00:50] TC: So, yeah, Essence VC, so we're a VC fund. We specialize in investing in developer tools and infrastructure. We back companies at the earliest stage so where a person that has a rough idea or a couple rough ideas, all the way do you have a team with the products? We basically are part of there to support companies and technical founders going after generally the enterprise infrastructure space.

[00:01:14] KP: So I hear people use terms like seed, pre-seed, A-round, where do you define yourselves in that spectrum?

[00:01:21] TC: Well, the definition of that term changes. I feel like I'm a reporter now. So, basically, I try to get involved as early as possible. Like a lot of companies, when they basically even have no clue what the product they're going to build is, I will actually back

them, maybe just an idea, but a very rough idea. So you can call that pre-seed, although pre-seed is getting larger. So I'm sure there're going to be new names for that. But really as early as possible, especially in the sense vertical that we're focusing in.

[00:01:50] KP: Gotcha. Well, my understanding is the earlier stage you are the more likely it is you might pivot. I don't know if that's for sure true. Maybe you have some data on it. Do you find that to be true? And if so, do you worry about that risk being an early stage investor?

[00:02:04] TC: Yeah, that's a risk that you can't really account for, but it happens all the time. I was a founder before. And actually although we didn't pivot, like completely change what we're doing, we had to try completely different products, maybe selling to different problems within the general space. So yeah, it's very common. Like you will start with some place and then build a product, or the early prototype, it didn't work out as much. You try out completely different side of the persona. Maybe it was the developer and now just becomes like an SRE, right? Maybe still in the rough large direction.

So every company's journey is so different. It's really, really hard to know maybe two companies we backed cross paths. And I know that would be tricky, but I feel like the only way you can do this is to get consents from founders. And if it does happen, then we just have to kind of trail the waters as carefully as we can. It's really hard, yeah. Early stage, anything can happen. That's definitely true. That's fun part of actually doing early stage, because you're helping the companies actually navigate the pivots.

[00:03:12] KP: Well, another property I see in early stage companies is, not everyone, but broadly speaking, they tend to be a little rougher on the edges in explaining their value proposition or describing how they're going to grow. Are there any things you need to do above and beyond to have a vision for who has potential and success at an early stage?

[00:03:31] TC: This is probably where I feel like I have been having so much conversations, and I start to realize I actually have an edge helping companies especially because I'm a technical background founder. I was a founder before. I've done so much in this infra space. How do I actually describe the vision and how to actually – And basically, a lot of

this has to be figured out because you're fundraising, right? You actually get investors excited. You need to have a story that's exciting for people that doesn't know your space as much, or know enough about the space but not really deeply.

And I've worked with lots of founders even last few investments where you can say they're rough on the edges. Like everybody have unique abilities somewhere. Maybe you have a very big technical background, or you have done some research, or you have a huge network. Somebody has to have some unique ability, right? But when it comes to like storytelling, that is actually not something people usually do that much. You probably didn't fundraise that much before, or never. It's a big jump, I realize, especially technical founders going after this space. They don't really know how to story tell or even know how to position themselves well. And that's one of the fun part doing this. I actually sit down with the founders, look at what they know, look at what they've done and what they're looking to do and completely rewrite their pitch deck over a couple times. Even come with new company names, or even come with different ways to talk about their products. And that has a 180 effect on their fundraising.

So like the rough edges is my advantage, I would say, like because this is where I feel I can help the most where not many investors can, right? So that it's true, rough edges are there. You kind of have to go past that. But you kind of have to find like what is really, really different about this team or person? And what is still missing? And what are potential markets this team can go into? And try to find is there enough to explore here.

[00:05:32] KP: Well, being a former technical founder and having an engineering background makes you somewhat unusual for a VC. I can definitely see where that would help you in shaping some of these early stage companies. Could you talk a little bit about your background there and how that's informed what you do today?

[00:05:48] TC: Yeah. If you ask most investors or VC, "How you got into VC?" Most people tell you're very accidental. I think, to me, it's really – Yeah, I never even knew what a VC is when I was engineer. I was software engineer, right? I graduated University of Washington, and I even CS, right? It was informatics. Learning a lot about programming and got into a lot of like open source myself. So I'm an engineer that really learned a lot myself, not

completely, but quite a lot to get into working on interesting problems as an engineer. I really interest in distributed systems and work on open source on the weekends and nights. And joined a couple early stage companies, one called Mesosphere, that got a lot of attention early days at 2014 because of Docker. I was able to work on pretty interesting problems. Mesos was powered in production in Twitter for nodes, Apple as well. And so I got to work with really smart people, really interesting problems. So that was kind of my background working on large-scale systems.

Then I became a founder myself. I left Mesosphere. I raised the seed round. I cofounded with two other cofounders and had to learn how to become from an engineer. I managed a tech engineer team, but I never been involved in marketing, or sales, or any of the other disciplines. Suddenly, start from fundraising, figure out how to talk about my company, all the way to actually making something happen. And I realized it was a much bigger jump that I was assuming before I even raised for my company. So that journey helped me quite a lot, because I had to become the CEO and really learn how to make things happen. How to get customers? How to convince them and do a lot of these kinds of things? And I don't think I did it really well even, right? I have a lot of learnings and failures in that. But that journey informed me quite a lot to now when I become an investor now. Really able to like understand what that journey might look like.

And so like, yeah, my journey into this investing side and how this technical background, I guess, it's really able to see how a person like me has to grow into something more larger. I can even feel like the emotional journey even to go through that. And so when I've helped companies, after I backed them, I try to have really as much empathy as possible to help them. So I don't say like technical ability, yes, it definitely helps me to navigate, right? Have the right network of engineers to help companies back in the space or understand how to position products and stuff like that. What I also feel like from a personal side, being an engineer to have to jump into it being a founder. That journey, I personally walk through it a little bit. And so I have so much more, yeah, empathy in general to do that.

[00:08:37] KP: Makes sense. Could we revisit some of the focus of your investment? I believe you mentioned dev tools and enterprise infrastructure as kind of being the broad

theme. Are there maybe any good examples in the portfolio of companies you want to highlight or just general features you're looking for in your investments?

[00:08:54] TC: Yeah. I really like companies that really targets just technical audiences. So it doesn't have to be software engineers. But if you're selling – Or part of your users are developers, data scientists, system engineers, or people are just more technical in general, I found to have a lot of sort of ability to understand and help shape the company to really get traction.

The companies that back are generally in that category. Either they are targeting a technical audiences, right? They're building software to help data scientists, help engineers be more productive, or ensure the quality of their applications, or like help automate some part of their jobs, or you have a lot of data and I want to really able to leverage that data, right? Find the best data, AI, machine learning and infrastructure to like make a difference around that data. So companies, I would say, that's in a data – There's big modern data stack movement. I have a couple companies in that space like Transform, Notable, some of those companies like that. Or open-source-based companies, because I've been personally involved in open-source quite some time. I'm really fond of companies in that space. This could be Encore, DBC and stuff like that. And companies are like helping data scientists, dev tools. Or companies even taking data and doing a bunch of automation like I mentioned. This is Alchemy. Yeah, lots of different examples. I'm just highlighting a few things on top of my head.

But yeah, kind of the themes are all around data and infra. And the common sort of help I bring is figuring out how to hire. Figuring how to actually sell and make the product better and know how to talk to the audience that you're talking to.

[00:10:37] KP: Are there any common things you see founders underestimating?

[00:10:42] TC: I think every stage has its unique challenges when it comes to the founding team. Most of the founders I back are mostly technical. And I think what founders usually don't have any experience in, they usually underestimate it. So engineers might think selling to companies is easy. Or if you're a sales or product person, you might think building

a product is easy. I will say it is not always true, but you either fully overestimate or you really underestimate. And I think most technical engineers really underestimate the marketing and the sell side. Especially a lot of engineers I talk to are like, “Hey, Tim, why are we really honing down on how we talk about a product? It doesn't really matter.” And I say, “Hey, it really matters, because how people talk about your product is the product when you don't have anything to use, right?” Even if you have a product, how people talk about you really matters a lot.

And it's really surprising, I think, for engineers. They don't understand how much work it takes to iterate and to really hone down that story. And this really ties into fundraising. Some engineers can fundraise really easily because of their backgrounds. Some, they couldn't. And the ones that couldn't, we have to really work through weeks of times, sometimes even months to get the pitch ready to be able to make that happen. And so I think this is underestimating things that you don't understand happens quite a lot. And how much time and shots you have to take to get to the goal? It’s not just one clear shot you get there, right? It's actually multiple shots. And so lots of things are really hard for someone never start a company to understand, yeah.

[00:12:27] KP: Good points. With regard to open source, someone not super familiar with it, like you are, might think the idea of open source is you know antithetical to having a commercial business. It'd be silly to invest in something that I can just go take for myself. What's the actual role of open source in modern startups?

[00:12:47] TC: Yeah, this is a question I feel like we've been really discussing even among investors, especially a lot of customers. Open source is interesting, because open source to a lot of people means different things. A lot of people feel like if I just put my code online in GitHub, I’m open source, right? It could be my own product. It could be part of my product. Yeah, like I said, if you just put your product out there with free license and free open source code, it will not help you. It will actually make your company much harder to run, because what will people pay for, right?

There're pros and cons of having open source, I believe. I think that open source is, in one way, yes, it makes it harder because you have to figure out what is going to be open source

and what's not going to be open source, right? And how do you actually make money? How do you make people willing to pay for something? Because, eventually, you have to be a business. You have to be a business and able to generate revenue. And is that going to be a service people pay for that you're just hosting it for you? Is this going to be other things out on top, like plugins? You have to figure that out.

But there's also a reason why we're actually very excited about open source, right? Open source is allowing companies especially in today's world where like developers, end users, are having a lot of influence on buying. Open source becomes this free marketing. Not to say free marketing, but it's like a very powerful marketing for users to get excited about something and able to contribute or able to actually gain trust and talk about it more openly in the internet for you. And so you start to have this more viral, people getting excited about a project in general. Versus, hey, if it's just a product, people may not be as excited.

It was interesting. As a developer, you have more inclination to talk about an open source project versus commercial product. And that amounts of excitement and people sharing and people talking about your project can really bring a lot more sort of like gravity into what you're trying to accomplish. And you can then capitalize on that. Hopefully become what they call the standard, which is really everybody think of X, they think of you, because you're the most popular or most faster growing open source project with so much people talking about.

Really, I feel like open source has a lot of benefits. It actually gets people to share and think of you as a more – Let's think about helping a company to market your products. You're thinking about something that's generally useful, and I'm just helping everybody to understand this, right? Because it's free to use. So, yeah, there's this really interesting thread to play. Because, yeah, something's harder and something's easier.

[00:15:31] KP: Makes sense. The data space and the infrastructure space as well have evolved quite a bit over the last several years. And I don't know that that's going to stop anytime soon. Are there any particular trends, things like no-code and serverless? Any things like that that are of particular interest to you as an investor?

[00:15:49] TC: Infrastructure, like I said, is evolving, right? I think if you look at like 10, 15 years ago, what infra investors are investing in are like networking, virtual machines, hard drives, things that power your machines sitting on your lap or in some room that needs to be have a temperature measurement in there, right? Now infrastructure is a lot more software. It's software-defined everything, software-defined network, storage and stuff.

And now I think infrastructure now going beyond, right? We're talking about serverless and no-code now, right? That's considered even sometimes infra or dev tools. Because the ability for somebody to able to build an abstraction and build your business around and really make your productivity goes much more faster. I think that's a goal, right? You're not just buying infra for no reason. You're really making infrastructure bets in your company so that all the rest of your businesses can go faster and better. And so it might be, "Hey, I'm relying on data and AI to power my business. I want to have a bunch of automatic recommendations to how I can run my business faster. How do I like reduce fraud or do a lot of decisions? I really rely on data in much more real time and much more places in my business." Then you need better infra to support that.

And so what I'm seeing a trend in infra is like you're really moving infrastructure to help more personas now, because what we used to think about infrastructure is just if I'm an infra company, I'm most likely selling to like system engineers babysitting a data center. Like seeing if things go down, go back, or monitoring, or something like that. Like I feel like infrastructure is expanding. I might be a business analyst, and I need infra because, business analysts are starting to write code per se. But this code might be drag and drop UIs. It might be like simple rules or low code, right? But things are powering users to do something on top. I actually consider all of that infrastructure.

And so when it comes to infra, we're seeing just a lot more abstractions. No longer just give me better network or storage, right? We're building more and more things on top. Like, now an infra could be, "Hey, give me a website I can drag anything on top of and becomes my personalization website for this particular persona." Infrastructure might be a way to automatically generate a certain number of ways to empower like my support person to able to write their own support ticket system, or even to like have quality measurements in different places. There's a lot of infrastructure that can be built, reimagine, that I'm actually

pretty excited about because I think there's a lot more people that are trying to be because of democratization, right? Empowering a lot more people to build their own apps, build their own workflows, or build their own insights from the data. And so whatever plumbing needs to happen – So I'm very excited about things that can enable things to happen. This could be serverless. This could be AI-generated, insights, or apps, or ways for people to actually gain faster way to query your data and gain really fast ways to visualize very large data sets, right? All these kind of things.

I guess I'm not saying very specifically, but a lot of these new patterns and new abstractions are being invented. And a lot of these, it will take time for them to come up. But I'm very excited about lots of these, because these can be pretty dramatic if it works. Yeah, so my headset right now is like what can we reinvent so that we can empower a lot more people to do things? And that kind of infrastructure actually is very interesting.

[00:19:38] KP: How do you typically find the startups that you go on to invest in?

[00:19:43] TC: Yeah. It's something – Like we call it sourcing in VC. And sourcing has been what I've been learning quite a bit, because I never worked on a VC fund, right? I was a founder. I raise money. Now I've done a little angel investing. When you start off, you always just start investing your friends, people that you most likely know or your friends of friends, right? People that heard you could be helpful for. That's where I started. And things started to change over time when I make more friends. This could be other VCs that I collaborate with. This is on the other angel investors I collaborate with. You sort of build your own network through just common co-investing.

And once you have a portfolio of founders, they become your spoke person. Like, "Hey, Tim is the most helpful person we have. Really talk to him." So our network of nodes just expand over time as long as they think highly of you basically. So yeah, the companies where I come from come from portfolio founders. They recommend me, other co-investors that we work with. I also have a podcast called Open Source Startup podcast where people might heard of a podcast. Or companies could come from anywhere. Actually, I had people reach out from GitHub. They saw my code contributions and it's like, "Hey, Tim, I saw you're an investor. I saw your code and actually, yeah, I wanted to talk to you because of

my GitHub account." And some in Clubhouse, people have reached out before. It really come from anywhere. Also, majority is like your own network of people basically, yeah.

[00:21:15] KP: Are you open to being pitched or do you prefer to control the inbound?

[00:21:20] TC: I prefer however we can find good companies. I have ways to outbound or basically finding companies I feel like could be interesting, or people that could be interesting. And I find them. I reach out. And also have inbound. So I don't have a preference. I would just say like it's really important to figure out how to find the right companies and people. And so like inbound, typically, people don't know you as much. We'll send you things maybe not infrastructure, right? Like it would be a consumer dating app. Those kind of things, I can't even invest in. So usually those will be just time consuming just replying like, "Hey, sorry, I don't invest in this area." So I don't have like a strong preference. It's just like it's harder to – It takes more time to sort of filter just because I have a very specific like area to invest in.

[00:22:04] KP: Yeah. Well, let's imagine there're two people listening on a speaker in a garage somewhere right now coding away on the next big thing in development tools. How prepared should they be or what steps should they take before they approach you?

[00:22:17] TC: Like I mentioned, I love talking to people and working with them when they still have just a rough idea. To me, I think you just really need to know what is the problem that you want to solve roughly, and also like what is the thought process that leads you to want to start something? That to me is like probably the prerequisite, because we can figure a lot of other things out. But sometimes if you're not so clear why you want to start a company, you just feel like it's a nice thing to do or it's a fun thing to do or it's a hot thing to do. It will be harder to get anyone excited, because starting a company is really hard in general. It's much tougher than people think. So the why matters a lot.

Yeah, once you have a why and also you have a problem you really want to solve and you think you could have solve it really well, even though you may not even know exactly, that's good enough already. I would love to talk to anyone in that nature just to figure out, "Okay, what are potential ways to do it? Maybe you have a research paper. Maybe you have an

open source project." I mean, the good thing about being technical, I can read through all that and discuss it with them. Figure out ways and re-figure out the pitches and all. I really love doing that. But it's going to be hard or maybe basically a waste of time if you're not really trying to do this seriously.

But a lot of things start off at exploration as well. So I'm generally very open to talk to people that's especially technical, really exploring a startup path. But I say like to people to get conviction, like really want to back somebody, they have to understand the why and the what's, right? And those two are very important.

[00:23:59] KP: When you get close to making an investment, it seems to me one of the considerations you have to have is, especially when investing in dev tools infrastructure, could one of the major cloud providers suddenly introduce this novel idea as like a basic feature? Not even necessarily maliciously, right? They could just do that, and, in theory, that's a threat that could wipe out whatever unique product or idea you're considering is. Do you have an approach to mitigating the risk that the idea would just be absorbed by a giant like GCP, Azure or AWS releasing it?

[00:24:33] TC: Everyone has the same consideration or problem especially in this space. We've learned quite a bit over time now, right? I think we're asking this question even like seven, ten years ago. It's definitely something we've been learning over time. Like what does Amazon do well? What does Google do well? What does Microsoft do well?"

And what I think what's interesting now is like not just the cloud providers can build stuff, right? We're going to see any large infrastructure companies like get into your area. Or we have Databricks now. We have Snowflake. We have a bunch of people fairly large doing a lot of different size investments. And so it's probably nothing new.

What we always try to figure out when it comes to infrastructure very specifically, right? Infra has a very unique space, because things takes time. And typically your revenue won't be – It's not like a SaaS product where like you can typically just see like very typical growth patterns and kind of invest there. You actually do need to understand the technology somewhat and understand the space and how things will shape out.

And so, yeah, that's part of a thesis of why when we back any company is to figure out like what makes this company able to own this category? If you're going to start, let's say, a low-code infra company, right? What is that they're doing that is just not the DNA of Amazon, or Google, or Microsoft? And it's just so hard for them to 180 what they're doing already to go for it, right? One way, they're not doing it because it's not obvious. It's a big risk for them, right? And so you kind of have to have something – One example, always be open source, right? You're not going to completely open search all EC2 or make everything just very easily portable from one cloud to another, for example, right? So somebody making that bet saying that we're going to build something that's completely portable from every cloud and also completely supported by the community.

You can argue that is a very viable way that, "Hey, Amazon don't like to just pour every workload to Google very easily." But somebody – If that can make that happen in that layer and that itself is very valuable, that is a potential way for you to make – Like, "Hey, Amazon probably won't do that." That also on its own is the assumption, right? So there're always assumptions when we're trying to make some investments. And I think we have to continue to test those assumptions, because things will change sometimes. Like sometimes we'll see some layers become just completely democratized. Like maybe the cloud doesn't care anymore. It's not a place we should compete. Let's just open source everything. That also creates some interesting challenges.

So as a startup, you're really just looking for people that can understand this and really able to move fast and can react – Not react. But just able to really change and adapt. Because, yeah, you never know. This market also changes quite rapidly.

[00:27:35] KP: A common path for startups is to go through several rounds of funding from wherever they start, to ABC, that kind of thing. You're investing in the earliest stages of that. With that in mind, where do you see is the vision? I guess you could make one investment and maybe that's all the startup needs to become a cash flow positive business. Or maybe the idea is just to bridge them to an A-round or something like that. What are the outcomes you're looking for?

[00:28:02] TC: Well, if you ask any investor doing in a space, you're all looking for the largest outcome. And I think my goal and, hopefully, what I'm doing well is to find companies that can get to the largest outcomes. And so, yeah, you definitely need quite a few rounds of investments. And what we see though in this current VC landscape is that, I think 2021, if you're an infra a company that found the right story and a right team, raising is probably not the hardest problem anymore. Like there's a lot of funds are really interested, especially like the largest growth funds, A-rounds funds, or even like let's say seed funds that are really large now, right? You've seen some announcement. They have lots of hundreds of millions of seed funds.

So funding is not easy, still hard in general. But I think if you can find the right entry into the market and have the right story to tell how to capture that market and have a convincing story to say, "This is how we're going to do it." Lots of funds can fund lots of rounds. And so my job as a small investor or pre-seed investor, it really is thrilling to help the team to iterate to a point to find that right story. And lots and lots of funds, we will fund the right companies or the companies that are getting the right story and traction.

Yeah, we all collaborate in this ecosystem that, "Hey, we'll back early companies and we'll help the later stage investors understand what this company is sort of – What their approach is different? Why they philosophically have a chance to own big markets?" And sort of calibrate there.

So yeah, we're also working with a lot of investors basically in this space to make a company successful. And we're definitely seeing there's a lot more capital available now. So I think, definitely, I'm lot less worried about, "Hey, how do you find money?" I think it's a lot more about like how does a team really able to execute, but also have the right story and angle in mind that will work in the long run so you don't feel like you're capped in small markets."

[00:30:16] KP: Well, for a company that's on the right track but maybe hasn't really found their story yet, you still need to do some sort of valuation if you're going to invest. How do you assign a value at a company like that?

[00:30:27] TC: Yeah, it's almost impossible. That's one thing when I started I have no idea. Still, it's really –I mean, the truthful answer, I think everybody will tell you, in reality what happens is supply-demand. Where companies, depending on how many interest there is, sometimes will dictate how the price will go. Because in the early stage, you cannot price companies easily, right? And every company you back hopefully will become the largest outcomes, right? You're not really trying to play, “Okay, this is–” I would say it's a 10 million valuation versus a 20. Of course, you'll make things different. But also if things will work out, it should work out in both cases. So it's a hard game. Valuation really happens so fast, because a lot of rounds happens quite fast now. And it's really a negotiation process between like the lead investors.

And so, yeah, as far as I can tell right now, the best way you can do this is to really figure out what is the comfortable range. And range can widely vary, but it cannot be too wide. This is what I believe we can make things happen, right? It cannot be just one number. This range would be good. So yeah, it's ongoing conversation challenge, because every company is so different. Some markets are more tougher. Some teams have more things in the background that can be proven. And also some teams just have a lot more interest in investing from VCs. And so all that gets into a factor. It's like, “This is a fair number just based on patterns we've seen.” So there's no really big science around it. It's actually really based on what's happening.

[00:32:07] KP: Well, I imagine, most technical founders will really focus heavily on the software they're building as sort of the key asset, which certainly that's part of what they're creating. But as you said, they need to think about the why of the business and a little bit about how they get to market and all those sorts of things as well. In reality, how important is the software?

[00:32:27] TC: It's a really good question. I think if you talk to lots of different people in this space, we always have good examples where the most powerful or most feature-rich, or most fastest database, or whatever, right? The software that can do something the best may not be the one that can win out the market. And we see that in different kinds of markets, right? We could say databases. We can talk about like CI/CDs or any kind of thing. What generally is true is, especially in this infra space, your software has to be

adopted and also be relied upon by the users. And not always the best sort of like fastest horse will win. You kind of have to build the software that allows people to understand, trust and like using it.

The experience of software also matters quite a lot. I mean, going back to the question, “How do you sell? How do you talk about your product and your software?” should be really one product. Like it's one thing. And the software is just reinforcing what you're talking about. Like if I say I’m the easiest to use developer SDK to build mobile apps, for example, right? Then it has to be easiest to use compared to any other option out there, right?

And so your software still matters because that's what people actually use. But if you say I’m going to be the easiest to use and then you're only a little bit easier, or it gets easier over after three weeks of learning something completely new, or something like that. Like the level of friction of your software also would impact quite a lot your value of your software of how people can quickly able to understand it. I think it's interesting. Yeah, you cannot just hone down into his very specific features and just say I’m the best software. You really have to figure out like what makes this particular kind of product in this market really tick and how do people really quickly, within a low amount of friction, able to understand and get excited about your product? And so software really reinforces that.

So yeah, it's interesting, because technical people typically think when they want to build the most complex thing, it's something you can brag about with other developers, basically. Like, I built this fastest thing. It relies on the latest whatever papers and whatever hardware, right? But then in reality, your markets, your customers either don't care or find that to be really hard to even understand. And then you can't sell, right? If people can’t understand, they won't buy. Software does matter, but not in a way that matters for engineers at least.

[00:35:22] KP: So for your investments, they must go from some stage from introduction all the way to when you cut a check. Can you talk a little bit about the timelines and milestones you're looking for when considering a company?

[00:35:34] TC: Yeah. And this varies quite a lot just because every company is quite different. When I'm doing, let's say, the teams that are still figuring out what they want to do, like a very, very early stage and haven't even know what particular market to go for. Those companies I typically work a little longer because I want to really get to know the person. I want to get to know – Or the team, the collection of people. What are they like, right? Can I work with this team really well that I feel like they receive my feedback really well? Or has really a good mentality and mindset going after this space or just this whole entrepreneurial journey overall. And can I able to tell what are the most unique abilities coming into this space? What can I do exceptionally well? That, I believe can show up in the product.

And so that takes time, because a lot of times like when we talk about earlier, rough in the edges. They could be pretty rough in the edges depending on how they are. But I don't just pass on them. For the ones I believe that has something interesting, I'll spend some time. Sometimes this will take like a month and a half or two months to really go through that. And it's like, "Okay, this is actually much more interesting than it shows on the surface." Then I can back them. So those take long.

Sometimes, hey, it's a company that has lots of things figured out, and open source is already out there, for example, or has already people using the product, or they built this before some other larger context and understands what exactly what they need to do and a lot of kind of things are in place, then it doesn't take that long, right? A couple chats, a couple of references and understanding. And maybe, is that a product? We'll try it and that kind of thing. Then it should be good enough to understand what the parameters are. So it varies quite a lot. It just depends on what is the state of the team and how much their story is clear. But I try to spend time on the ones I believe have a good shot to have a good story.

[00:37:36] KP: With regard to the team, I guess the need is going to vary a lot by what the product or service is. But I'm curious if you have any common recipes for success, I suspect five technical co-founders might not do as well as a more diverse group. What does it really take to get a good company going?

[00:37:53] TC: Yeah. I guess when it comes to companies, I found many variations that has worked and many variation has not worked. And I don't really have a good – Like maybe put it the other way around. Like there are teams – The three technical co-founders could work. And I backed quite a few of them myself. One thing I always believe is that everybody can learn like, “Hey, I wasn't a CEO before. I was able to learn just enough to get somewhere.” And you really are looking for people that can learn, that has the mindset, that can receive feedback, that has the right mentality to understand what it means to learn something new and not rely on what they know so much.

To me, I’m looking for people that has the right mentality, has the right mindsets. And I really like people that can have confidence in what they have done so well, but has enough really good amount of humility of what they don't know and willing to really partner and for people to really help them.

So yeah, like to me, recipes assess are not what they know really well. Because sometimes when you find the greatest marketing person, the greatest product person, the greatest technical person, however you measure it, you smash them together and they don't work because they rely on too much what they know and they don't adapt well to the market. Because most of the times, as a founder, you are always learning a whole lot. Like when you start at the pitch level when you just raise your round, you really only know like a tiny fraction of what you need to know to succeed. And this happens to every single company. So it's like playing your RTS game, right? Your map is completely blacked out. You think you know every single thing, but you really in reality don't know anything. Not enough to really succeed.

And so the teams that can really work well are the teams are willing to know that, “Hey, my mental model of the map is not just all I know. Actually, there's a huge compass of things I need to go figure out. And I’m willing to learn and willing to take that on.” Those teams succeed much faster, because they are able to put their foot down. They don't wander around. They execute. Well, I’m saying that the teams that can execute really well are the ones that willing to try things, willing to actually take a stance really quickly, but also really to take feedback and readjust really quickly as well. Those perform just much better in general.

[00:40:24] KP: Well, good insights. Tim, where can listeners learn more about Essence VC online?

[00:40:30] TC: Yeah. So I have a website, essencevc.fund. Have a little bit of information. I'm going to release more information about the fun over time. I do have a podcast as well and people will listen to me and Robbie from Cowboy. We're talking to a bunch of open source founders about other early learnings and challenges. So yeah, follow my Twitter, follow the podcast and definitely have more information out there. But as much as possible, I've I told everybody that is interested for my fund or even people that are considering taking our investments, I always tell them to go talk to the founders we back. We have the company logos on our website. Talk to them you know. They are the source of truth, right? They're the people that actually work with me and they have the best like knowledge of what does it look like to work with me in general. But yeah, those are places that I have information on.

[00:41:23] KP: And remind listeners the name of the podcast.

[00:41:25] TC: Yeah. It's called Open Source Startup podcast.

[00:41:28] KP: Sounds good. Tim, thank you so much for taking the time to come on Software Engineering Daily.

[00:41:33] TC: Yeah, it's a pleasure. It's great to be here.

[END]

