EPISODE 922

[INTRODUCTION]

[00:00:00] JM: Indie Hackers is a platform for independent software businesses to discuss strategy and find inspiration.

Courtland Allen founded Indie Hackers with the goal of sharing the stories of these businesses, and the company has become a thriving community of entrepreneurs, and engineers, and creators. Business is a creative medium, and the definition of a successful business is as subjective as the rules for what makes a successful work of art. A business owner can be miserable running a company that generates millions of dollars a year, and a new entrepreneur can feel ecstatic from making their first $5 sale.

Indie Hackers is a platform that is impossible to define in relation to things that you've seen before. It's a media company with a podcast that most Software Engineering Daily listeners would probably enjoy, and it's a social platform for learning how modern software companies are built. It's a place where makers post their own progress on their creative projects. I've posted mine on Indie Hackers, and it's a pretty cool magazine of businesses and things that are thriving.

Courtland was on the show three years ago to discuss the Indie Hackers Movement in its nascent stages, and he returns to the show to discuss the thriving platform as it exists today. We had a great wide- ranging conversation about software, game theory and podcasting.

We are hiring a head of growth for Software Engineering Daily. If you like Software Engineering Daily and you consider yourself competent in sales marketing and strategy, send me an email, jeff@softwareengineeringdaily.com.

[SPONSOR MESSAGE]

[00:01:49] JM: Logi Analytics is an embedded business intelligence tool that allows you to make dashboards and reports embedded in your applications. Create, deploy and constantly improve

your analytic applications that engage users and drive revenue. You focus on building at the best applications for your users while Logi gets you there faster and keeps you competitive.

Logi Analytics is used by over 1,800 teams, including Verizon, Cisco, GoDaddy and J.P. Morgan Chase. Check it out by going to L-O-G-Ianalytics.com/datascience. That's logianalytics.com/datascience.

Logi can be used to maintain your brand while keeping a consistent, familiar and branded user interface so that your users don't feel like they're out of place. It's an embedded analytics tool. You can extend your application with advanced APIs, you can create custom experiences for all your users and you can deliver a platform that's tailored to meet specific customer needs, and you could do all that with Logi Analytics. Logianalytics.com/datascience to find out more.

Thank you to Logi Analytics.

[INTERVIEW]

[00:03:13] JM: Courtland, welcome back to the show.

[00:03:16] CA: Thanks for having me, Jeff.

[00:03:16] JM: Our first show was three years ago, believe it or not.

[00:03:21] CA: That's crazy.

[00:03:22] JM: How have things changed since then?

[00:03:24] CA: Well, when we first recorded, I didn't have a podcast. I think as a direct result of coming on this show and talking to you about podcasting, I think after recording, we just talked about podcasting for a long time and people have been suggesting that I start a podcast for Indie Hackers, because at that point in time it was just a website. It's basically a glorified blog and people were like, "Hey, I like the interviews you're doing online. Can you just do a podcast so I can listen on my way at work or in the kitchen?" I'm like, "No. That sounds like a ton of work,

and I would never do it," and you pushed me over the edge and convinced me that, yes, it is a ton of work, but it's super rewarding and it's kind of the future of content. So why wouldn't I have a podcast? So I think that was – What? Like November of 2016, maybe even October?

I launched my podcast with three episodes in February of the next year, 2017. The podcast is 2-1/2 years old now. It's getting crazy number of downloads. It's by far bigger than the website. That sort of help bootstrap the podcast in the first place. So that's one major difference. What else is different? Indie Hackers is now owned by Stripe.

So when I first came on your show, Indie Hackers was a small bootstrapped one-person business and I was making – I don't know, a couple grand a month in advertising revenue and trying to figure it out. IN April of the following year, I got an email from Patrick Colison, the CEO of Stripe, who asked if he could acquire Indie Hackers.

So we met over brunch and talked about it and we run some numbers and the next day I said yes. A month later I was part of the team at Stripe fulltime working alongside my brother. So I've been at Stripe now for almost 2-1/2 years. Far longer than I had run Indie Hackers on my own. It was only like nine months from founding to acquisition. So those are probably the two biggest differences.

[00:05:00] JM: Now, the situation that you're in, you're taking what started as a media product and turning it into a larger platform. I've tried to do the same thing a little bit with Software Daily, Software Engineering Daily and Software Daily. It's been quite hard. At this point I don't really know if I am a straight up media company. I don't know if I'm aiming for a media company that's supplemented by technology. Can you give me some advice on how to go from a media company to a platform?

[00:05:37] CA: I can't give you general-purpose advice, but I can tell you what I did and maybe try to like reverse engineer, like extract some lessons of the benefit of hindsight. But it's really hard to say if those lessons would work for you. For me, Indie Hackers is very much still a media business. I mean, we still produce a ton of content. We still have the podcast obviously, and that's all media.

But at the same time, we've got this platform community type thing going on that was part of the vision from the very beginning. That is content which is user-generated content. That's a bunch of people on the community forum asking each other questions and sharing tips with each other and helping each other out. If I were to get a buy bust tomorrow, those people would still be there talking to each other. So I think that's the core of any sort of community, is people who are empowered and able to help each other out.

I think at the end of the day, there are certain things that people are willing to do that with and there are certain things that they're not. For what I'm working on Indie Hackers, it's all about people starting Internet businesses. It's all about people helping each other overcome these challenges. There is an almost inexhaustible number of conversations and content that people can generate and talk about in that field.

People constantly feel lonely. They constantly feel frustrated. They constantly need help. There are there for the long-term. They build up a profile. They invest. They want to share what they're doing and ask others for help. When they learn enough, they want to soft of flip the script and help others who are below them.

In some ways, I don't want to say it's easy, but it's easier than many other fields where I've seen people try to start communities, because let's take software engineering for example. Can you build a community around software engineering? Of course. There are lots of people who've built communities on software engineering, but the conversations are going to be different.

So Stack Overflow, for example, you could call that a community. It's certainly a community. But probably the default way of using Stack Overflow was to sort of ask a drive-by question, "Hey, I've got this issue. I need it resolved. Let me ask a question, because I've exhausted every other resource on the Internet." If you get a response, great, you go away for another year until you have another question and you can find the answer on your own. You're not really an invested member of the community.

There are some people who are like super Stack Overflow users who answer tons of questions and they're really bought in to the gamification and really like helping. But that's not I would say the majority use case of the site. So I think it really depends what market you're in. What

problem you're solving. What you want your community to be based around. Just like the essence of a he normal average conversation sort of dictates how easy it is to pull the community and how easy is it to get it started.

Indie Hackers, like I just followed the Reddit playbook. I've created a forum. Yeah, I made a bunch of fake threads. I made a bunch of fake accounts. I talked myself a lot, then I would send those conversations out over email. Every now and then I would trap a real person into a conversation with like three of me. After enough time, there are two people talking to a bunch of me. Then there are three people talking to a bunch of me.

I think every community looks somewhat like this to begin with. It's like the creator of the community doing something that doesn't scale. Something that's a ridiculous amount of work that's not sustainable just to recruit people and keep them engaged and make it a great experience for them. Then eventually at some point, flip scripts, the script flips and everybody is sort of sustaining themselves. But it takes a long time to get there. Even with the Indie Hackers community, I still put in a significant amount of effort to make sure the community is a great place. We have a community manage, and you, in my opinion, should never really stop doing that. It just becomes easier over time.

Probably the last tip that I would give is I think you want to start small. With any sort of community, you basically want to shrink both time and space in such a way that your community feels active and lively, because any sort of community is just based around social connections. If people feel that the social connection is not there and it's not active, they're going to immediately assess this community is not valuable and they're not going to come back.

So I will post on Stack Overflow, because I feel like it's active and I'm going to get a response. If I don't think I'm going to get a response, why would I be part of that community? I'll show up to like a running group on Saturdays because I feel like other people are going to show up. If it was just me by myself or not enough people for me to make friends, then why would I show up to that running group as supposed to just running by myself?

So you want to restrict the size of your community sort of the space. That will be something like if you're going to have a party, you have it in a very small room. If only 10 people are going to

come, don't have a party in like an auditorium if you're going to only invite 10 people, because it's going to feel dead, right? Whereas if you had a party here, like it will feel great with 10 or 15 people. So that's kind of the temporal component, or that's spatial component of shrinking it.

Online what that looks like is – I don't know. Instead of starting like a gigantic forum with like 10 different sections on your disk course forum, just have one section on this disk course forum where there's not that much room for people to post in all sorts of different places. Otherwise, every place is going to look dead, right? Just start something really small. Let that get to the point where it feels crowded, then overflow into like a second category page or something.

There's a temporal component, which is sort of restricting the time at which your community exists. One difficult thing with doing that online is people tend to start forums and chat rooms. These things are always online. They always exist.

So you're really like greatly multiplying the amount of effort and energy that needs to go in to making the making that community not feel dead. Whereas offline communities usually are very temporarily restricted. They'd be like, "We have a meeting time and a place. We're going to meet on the third Saturday. We're going to have poker on Wednesday night at six." It doesn't matter if no one's playing poker at any other time. If they're all there at Wednesday at six, it feels like a lively thing.

So I've seen communities start online where people were like, "Okay. Instead of starting an online forum or a message board or something, what we're going to do is we're going to post one question every Friday and everyone who is in the community is going to come answer this question and discuss it." Then the community doesn't exist outside of that.

So everyone sort of thinks, "Oh, this community is thriving," even though there only has like 15 or 20 people, because it feels like it's thriving, because it's been temporarily and spatially restricted set. I think that's sort of the best hack to sort of get a community off the ground. From there you just like slowly grow and put in lots of effort to get it to be bigger.

[00:11:12] JM: So one of the themes of your podcast and building an online business in general is when do you delegate and how do you delegate? I've had some conversations with you over

the years about as you are taking this media thing/platform thing and you really enjoy doing it by yourself and you've got some traction by yourself where you start to identify, “Maybe I should be outsourcing this thing. Maybe I should be delegating this thing. Maybe I should be working with somebody else on this thing.”

Half the time it feels like that could be great. But the other half the time you’re like, “Then I have to spend so much time managing this person. I’d have less time for mobile chess and less time for reading random books.” Part of the value of the Indie Hacker idea is you have total independence. Basically, the only employees you have are SaaS products and for many of the companies that you talk to.

Have you found that delegating saves you time or you end up spending more time on things like management issues. What’s the tradeoff there?

[00:12:24] CA: So I went into this whole Indie Hacker business very much on the side of I want to work alone. I had almost always worked alone basically in my room as a freelance web developer or on my own startups. I didn’t like the idea of having to manage people.

It sounded like swamp that you could easily get stuck in and lots of unpleasant things could happen and it might not necessarily make you more efficient. I didn’t like the idea of this business that I’ve created on my own suddenly feeling like a real job, because I have the responsibility of someone else's paycheck and someone else is coming to work. If I want to not work for a week, how can I tell all my employee not work for a week or that I’m going to disappear for a week, but they have to keep going. It just felt like there are lots of downsides to doing this.

I've since changed my position on this. I think that I have a lot of basically fears that weren’t founded in any real experience. It was just fear of the unknown. I'd never done this before, and therefore I assumed a lot of bad things that aren’t necessarily true. Through talking to so many different founders, I've heard so many different perspectives on experiences that people have had with hiring.

I was talking to a friend who said she will never hire again, because she hired two people on they're both terrible hires. They did exactly what you said. They drained all of her energy and her time. She spent so much time managing them and trying to help them and they were messing things up for her. She moved marginally faster in some ways, but so much slower in other ways and she's just done.

I've made bad hires myself, but I've also made some really amazing hires who were like a godsend. Who, for example, our community manager with Indie Hackers, Rosie Sherry. I basically don't tell her anything. She just does her things. She just knows what to do. The form will be much better. There'll be all sorts of connections being made.

She's reaching out to people, getting them to participate. Curating a thread, she's cleaning up spam. She's like taking hold of our Twitter account in the last month because it seems like a fun project for her and she's like grown our Twitter impressions by like three-X. She's like putting together other – She's just doing all sorts of crazy stuff like I don't necessarily have to oversee and I can just trust that she does a good job, because she does.

[00:14:18] JM: She's proactive.

[00:14:19] CA: She's proactive. I feel like I've hired another one of me. It's just like I cyclone myself and, of course, trust another one of me to do good things, because that's another one of me. But she's another one of me who also brings a different perspective. So she's doing good things that another one of me would never do.

So I'm now bought into this whole Silicon Valley mindset of like hire the best people. Hire well. I've heard that for so many years. I was like, "Yeah, of course, you want to hire well." But like the difference between hiring someone who's and hiring someone who's just sort of average and is going to need a lot of management a lot of handholding is staggering. I mean, you could easily double or triple your effectiveness by hiring one great person.

[00:14:55] JM: Yeah. You've tried a lot of different things within the Indie Hackers platform, and sometimes when I look at Indie Hackers, I think about Quora, because Quora has tried a lot of different things in a singular platform, and the things that don't work they rebranded, or they

throw it to the side, of the phase it out or whatever. They're pretty good at trying stuff and then phasing it out. I guess Facebook is kind of like this too. Kind of a one person development team or certainly a smaller development team though.

What's your philosophy around experimentation?

[00:15:30] CA: Do a lot of experimentation. Yeah, I am a one person developer team, but I'm going to be hiring soon. I'm very excited about that. It's pretty hard to build a social network by yourself. I think you have to experiment. At the end of the day especially if you're building like some sort of consumer app, something that's social, there's all sorts of second-order effects, unintended consequences on the things that you do that you can't predict. So even things that you don't intend to be experiments end up being experiments, because they don't work out.

There have been plenty of times, for example, earlier this year, I tried to sort of rescan Indie Hackers and make it work with more Twitter-like mechanics. So instead of being forum-like where there these discussions you have to click into. It was more of like a feed and you can sort of just scroll and read things, and people did not like it. It accomplished everything in my checklist, like had people making more posts? Those posts are getting more responses? But then there are all these like weird second-order effects it doesn't intend, where it's like instead of one post on the forum getting like 100 responses and everybody comes to the website seeing that at the top and saying, "Oh, this site is lively and active. Look, this post has hundred responses."

Now, the responses are spread across more different posts. So every post has like seven responses. I'm like, "Oh, that's exactly what I want. Now, everybody's getting responses." People who come to the site are like, "Oh, this place feels pretty dead. I'm not going to come." So then the traffic drops and I'm like, "Oh, I had no idea that would happen." So that's an experiment. I tried it out. I need to revert it. Sometimes you don't even know why an experiment failed.

There are some things that I've built that I just built out of order. So one of the first things that I've built after joining Stripe was this giant directory of products. So you could basically say, "Hey, I'm Jeff. I'm working on Software Engineering Daily. Let me create a page for my product."

You have this cool timeline that I designed where you can like basically post updates to your timeline. I was like, “This would be so cool if people had this.” It is cool. People do have it, and I think it is great.

But at the time, I had no way to connect that to the rest of the site. It was just like a product design faux pas on my part, or people didn't really want to use the directory and browse products and read these updates. I just kind of it languish for like a year and a half while I was working on other things. Then finally, maybe four or five months ago, found a good way to connect it back to the rest of the website. Now it's like the fastest growing part of the website. It’s doing really well. People are posting all sorts of updates to their timelines and people are commenting and liking on it.

[00:17:36] JM: That’s milestones.

[00:17:37] CA: Yeah, that's the milestones [inaudible 00:17:37]. Exactly.

[00:17:39] JM: People could see my stuff.

[00:17:40] CA: Yeah, people could see your stuff. If you go back and look at the older products, like some people would post 10 things and get no likes.

[00:17:44] JM: That’s a great product.

[00:17:46] CA: It works really well. I’ve gotten better I think as a product designer overtime at understanding like what people want and how they’ll behave, but it's definitely been a lot of trial and error to get there.

[00:17:56] JM: Why didn’t that product work initially?

[00:17:59] CA: So if you think about the way that – Maybe the best way to explain it is by explaining what does work. So the milestones feature on Indie Hackers is kind of a leaderboard that sits at the top of the forum. It’s almost like Product Hut where it resets every day, and

anyone who owns a product page can post a milestone, which is just sort of an achievement they've made to their timeline. Exactly.

So you might say, “Hey, I just reached the million download mark or I just got my hundredth sign up or something, or I just made my first dollar from a paying customer.” People love reading the milestones leaderboard and saying, “Oh, this is a really cool accomplishment. Let me tell this person congratulations, or are I want to ask how they did it, because I also want to land a sponsor,” or whatever the milestone is.

[00:18:33] JM: Without any traffic, you didn't have incentive for people to post their milestones, because there’s now automatic distribution?

[00:18:39] CA: I didn't have the readers. So I had people posting, and they would post milestones. There just wasn't like a milestone leaderboard where people could go to read this. You don’t have a good way to browse it. So you would have to, if you wanted to read what somebody was posting to their product page, go to the product directory, search for some product that you like. You might say, “I want to find somebody who’s working on something, in e-commerce that’s making at least $10,000 a month.” Then it would filter. You’d click that and you would have to read their individual updates.

So most people's updates weren’t getting any likes or comments, because very few people were doing this. Now, it doesn't matter who you are. All of your updates get piped to a single feed. That feed is visible on the homepage. So everybody who visits Indie Hackers will see. Now your posts have some degree of visibility. So I think that was a sort of just a product design faux pas on my partner. I had some good ideas, but this is a couple years ago. I think it took me a lot of trial and error and other experiments before I really understood how to take these ideas and wrap them in a package that the actual user base would like.

There’s a lot of that went into that. For example, the leaderboard mechanic is really important. That's one of the biggest things I learned from my attempt at redesigning the website to look like Twitter. People actually don't want you to have a website that's very egalitarian. They don't want all the likes and comments to be spread out evenly over all the different content. If you’re Stack

Overflow, let me go back to that example, or maybe even Quora, that's kind of fine, because most of your traffic is coming from search.

People are just sort of doing these drive-by searches. They see an answer, they leave. But if you're more of a dedicated community where regulars hang out all the time, at the end of the day, the best posts need to get the most attention. The best milestones need to get the most attention. The really low quality posts, even though it feels bad to me as a creator to see them not get responses, you probably don't want anyone to really see those. So it's better to sort of have a leaderboard format, have the best post sort of rise at the top and stick there like you see on Reddit or Hacker News than it is to sort of spread out the love.

I'm constantly wrangling with this problem of like, well, thousands of people are posting milestones, but not thousands of people can get responses, because they don't have time or the space, excuse me, to show all those milestones on the homepage. How can I do that effectively? The leaderboard mechanic is, "Just show the best ones and have everybody sort of compete to get to the top and don't feel so bad that not everybody wins, because everybody sort of –" There's a whole bunch of psychology that comes into why it works and why people were okay with it.

Generally speaking, people in any social setting will tend to copy what they see working. If you live in like a tribe or something and you see Mary is cracking opening coconuts with like a rock that she's found. You're going to be like, "I want to crack open coconuts. That seems to work." If you're on an Internet forum and you see someone getting to the top of the leaderboard because they're being helpful and they're explaining their post rather than just posting something and leaving, then you'll say, "Oh! I'm going to be helpful and explain what I do, because that seems to be working for them."

So like this whole leaderboard mechanic has all these advantages that are super helpful that I now understand a lot better that beforehand I didn't understand at all. So it's like just through experimentation and constantly tweaking and building the site, I'm learning all these different stuff about product development and how to build a social app, that in the beginning I had no idea about and I was sort of just guessing and checking.

[SPONSOR MESSAGE]

[00:21:46] JM: Today's episode is sponsored by Datadog, a cloud scale monitoring service that provides comprehensive visibility into cloud hybrid and multi-cloud environments with over 250 integrations. Datadog unifies your metrics, your logs and your distributed request traces in one platform so that you can investigate and troubleshoot issues across every layer of your stack.

Use Datadog's rich customizable dashboards and algorithmic alerts to ensure redundancy across multi-cloud deployments and monitor cloud migrations in real-time. Start a free trial today and Datadog will send you a t-shirt. You can visit softwareengineeringdaily.com/datadog for more details. That's softwareengineeringdaily.com/datadog and you will get a free t-shirt for trying out Datadog.

Thanks to Datadog for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:22:47] JM: Do you feel like you're creative identity is tied to Indie Hackers or do you want to expand as a creator or an artist? Are there other mediums that you, Courtland Allen, want to get into?

[00:23:04] CA: Totally. I watch a lot of anime, Jeff. I love anime. I love storytelling in general. I love movies. I love great TVs. I'm kind of a snob. I don't want to watch mediocre stuff. I don't like watching bad stuff. I find myself when I'm watching television or movies or anime just critiquing like little storytelling choices that I just think are efficient or like not good or not as engaging or as good as it could be.

So a part of me is like what right do I have to sit around critiquing stuff if I've never written a story in my life? So a huge part of me is like would love to tackle the challenge of being a storyteller, being a writer, being creator at some point my life. On the flipside, I think there's a lot of wisdom to sticking to your area of expertise. I put in so many years into learning how to code, learning how to design, learning how to market and sell, learning how to build apps. I've built up

a pretty like significant skillset there. To just toss that all aside and walk into to some other field and be totally green might be unwise, but at the same time, it would be pretty fun.

[00:24:01] JM: So when is that going to happen?

[00:24:02] CA: When is that going to happen? Probably not –

[00:24:06] JM: Not anytime soon.

[00:24:07] CA: Not until I'm done with the Indie Hackers phase of my life, which I don't see ending in the foreseeable future. The flipside of all of this is that if you –

[00:24:14] JM: You got to have a hobby.

[00:24:15] CA: I have hobbies, but my hobbies and more so like they're soothing hobbies. I don't know if my hobby can be just as mentally taxing and just as much work as my full-time job. At the end of the day, coding and building an app is pretty taxing. Being a writer, honestly, is not that different. There's not that much variety. If you're a writer, you're probably huddled over your computer all day the same way that I am. So I don't want my hobby to be the same thing.

The hobbies that I actually have now are like cooking. So at least I go to my kitchen. You mention playing chess on my phone. So I'll go on long walks and play chess or hang out with friends and traveling and going camping in a of couple weeks in the Guadalupe Mountains. So those are hobbies that are very different than my full-time job. I think if I ever switch over to writing, that will be like a full-time thing. There will be nothing else that's my main focus.

The other thing to think about is that there are a ton of advantages that you can accrue by sticking with one thing and sort of take into the next level.

So one of the people that I interviewed on the podcast is Chris Savage of Wistia, and he's talking about in his early days how they're coming up with the idea for their business and how people would come to them and be like, "Hey, you guys are the video guys. You know a lot about video. Help me with this."

The reason that kept happening, they kept having opportunity present itself to them is because they had established their reputation as being the video guys, because they were the at best video and sort of the social circles that they ran at and no one else really knew anything about video in 2004 compared to what they knew. When I think a lot about Indie Hackers and the advantages that it has, there aren't really a lot of competing social networks for founders out there. There certainly aren't any as big as Indie Hackers. When you are kind of at the top of anything, it's the leaderboard effect, basically. You get an outsized amount of attention, almost an undeserved amount of attention and focus and opportunities.

So I keep thinking about how far can that really go. What is the craziest future that it looks like if Indie Hackers does well on every front? Even when I joined Stipe, I think my vision for Indie Hackers was I want to be in Indie Hacker myself. It is something that will pay my rent. That will give me the freedom to – I don't know, live where I want and work whenever I want. I was a tried and true Indie Hackers and I joined Stripe. Patrick was like, "How big can this be? Can you have 200,000 people visiting this every single day?" I'd literally never thought like that. It hadn't even crossed my mind a single time.

Then I started crunching the numbers and thinking about a strategy and realized like, "It's actually possible." If I think this thing is good for the world, then it's hard to make the argument that I shouldn't try to bring it to more people and shouldn't try to make it bigger while also keeping it at the same quality or better.

Then I started realizing like all the different advantages that could accrue if Indie Hackers is super big, right? It's like meaningful for me. It's meaningful for the people who are part of it. The community is more helpful basically the bigger it gets, because, for example, people who are more skilled at business come in, investors come in. People help each other out more. It's like more of a thing with a capital T. So it's really hard to get away from.

I feel like one snowballs rolling down the hill and getting bigger as it goes. At some point, it's like, "Oh, it's big enough. I'll just walk away." But I don't think that's going to happen anytime soon. I think it could be much bigger, much better, and probably the more things I build, the

more things I add and the better things, get the more ideas I have for how Indie Hackers should grow and change overtime. So it's really hard to get away from that.

[00:27:22] JM: That's the advantage of being in a large and growing market.

[00:27:25] CA: It really is.

[00:27:27] JM: I think picking the right market is really one of those durable pieces of advice even if you build the wrong product in the right market. That's many times better than building the right product in a wrong market.

[00:27:40] CA: I have a whole order of operations for how you should pick what you want to work on. The first thing for me is founder product fit, which is become a more discussed term of late. I think no matter what you work on as a founder, you should always work on something that like you will enjoy working on. Because if you don't, you're going to be up against a bunch of people who do enjoy working on that thing.

If you want to get up and build a better Indie Hackers than me today, than you're going to have to literally wake up every day and put in like hours and hours and hours and days and days and weeks and months of work happily doing it, because that's what I'm doing. If you don't have that, then like it is a matter what strategies or tricks you follow. You're not going to be doing like the very basics.

So I think that should be like the number one thing you do as a founder, figure out what you actually will enjoy doing day-to-day. But then after that, market. If you're not in the right market, if you're serving like some sort of very shrinking segment of people or there's too much competition or it's just like there's so many different variables with the market. But if you're in the wrong market, it's like you're just fighting an uphill battle and it's going to feel like a slog every day. Even if you love it, pretty soon you won't love it, because it's not going to be working out for you.

[00:28:41] JM: I do feel like some of the people you've interviewed on your show – By the way, I seriously listen to your show. I don't know, but not in every episode. Certainly, yeah, what we're

talking about with if you publish more episodes, you would just be cannibalizing my listens. I think I am hitting my rate limit for how much Indie Hackers podcast content I will consume.

[00:29:01] CA: Two a week.

[00:29:02] CA: Two a week, yeah, roughly. I mean, sometimes going to the back catalogs because there are some episodes I've skipped, but it's something I want to listen to on a regular basis. I definitely would need my episodic dosage of Indie Hackers. It's totally distinct. But I do feel like there are some people who you've interviewed on the show who don't really care about the market that they're in.

But they fall in love with just the idea of running a business and the autonomy of it and the operations of it. I mean, it's certainly something that surprised me about Software Engineering Daily is there are so many like stupid operational things. Initially I'm like, "This is a big inconvenience." Then overtime I'm like, "Why do I like doing this?"

[00:29:44] CA: They're addicted to the operations.

[00:29:46] JM: It's not necessarily the operations, but it's just the fact that there is this weird engine that I've built and I'm kind of like, "Yeah, I'm kind of into this," and dealing with the minutia of it becomes fun.

[00:29:57] CA: Building a business, it's almost like a programming project, right? Where you work on something for long enough, there's like all these different classes and modules that you built and all sorts of different little levers you can pool and knobs you can turn. It's just something fun about holding on that complexity in your head and keeping this machine running.

[00:30:15] JM: Anything about how the sausage is made? Some random element of Indie Hackers that is one of those pieces of minutia that people might not expect?

[00:30:24] CA: Oh God! I'm like the opposite. I don't like the minutia. Any sort of craft that's not permanently pushing the business forward I look at as like a drag on my velocity in my trajectory and where things can go. For example, I'm really bad at checking my email. I'm deliberately bad

at checking my email, and I feel bad when I check my email, because there are people in there who I haven't responded to for a week and they had something that was somewhat urgent, but I just don't want to spend four or five hours a week checking email when I could spend one hour a week checking email and those extra four hours like writing code for the site.

So I love writing code, but I wouldn't consider writing code minutia, because whenever I build a feature, that thing is there and in perpetuity forever helping people out on the website. The podcast is another thing where it feels like sometimes really fun to record, but at the same time, it's an endless slog, because no matter how many episodes I release, no matter how good an episode is, by next week it's basically a chopped litter and I've got to record another episode, right? Even if I do a better job recording episodes, it just raises the bar and the expectations that people have and that kind of feels like a treadmill that I can never get off of.

For me, I don't like the operations. I'm constantly thinking about how can I hire someone to do this operation for me. How can I get someone else to title the podcast episodes or add the show notes or write the descriptions? Can I hire somebody to check my email for me? Can I hire someone to do all the tiny little friction things so I can just do the thing that I think has the most value?

[00:31:45] JM: This is so perplexing to me that you say that, because how is writing code the best use of your time? I mean –

[00:31:50] CA: It's not. That's why I'm hiring a developer as well.

[00:31:53] JM: Oh, okay. Are you going to become the chief design officer, or CPO, something?

[00:31:58] CA: Indie Hackers is not nearly big enough where it's not valuable for me to write code, especially being like with the particular skillset that I have, I can write code faster than a lot of people. It's just like I can do the design and like the sort feature mechanics and UX and do the backend and the frontend pretty quickly. It doesn't need to be like a meeting of like six different people to come together to do this. So it's pretty efficient for me to write code.

I also know the code base since I wrote it. So that's pretty efficient. But like, theoretically, there's some point in the future where I have like six or seven developers and it's like I'm primarily better off just like designing the feature or something. Then at some point I could hire a designer or a product person and they could design feature, etc.

I think early on in a business's lifecycle, usually the founder should be the product person. If you don't have any cofounders who can code, you probably need to be the person writing code.

[00:32:49] JM: I mean, I agree with you, but software engineering is pretty well defined, right? You could be – I mean, I guess there are a lot of finer touches to Indie Hackers. That's what I'll say, is like there are all these little edges that you've sanded in very pleasing ways.

[00:33:08] CA: I mean, a lot of it is experimental.

[00:33:11] JM: Give little animations and things.

[00:33:13] CA: People frequently request that I build a mobile app. I have no plans to build a mobile app anytime soon, because I just think, "Well, that would basically drag my productivity into the mud, because now every time I build one thing, I have to build it twice." But most importantly, the website is basically a huge experiment, right? Is it a forum? Is that a media company? What is it?

[00:33:29] JM: Wait, but you're always thing about mobile experience. Why don't you just have like it's a React Native basically or like –

[00:33:36] CA: It's just friction, right? Why don't I just like have the responsive design, do the bare minimum and then I'm free to move as fast as I can iterating on the website? For example, with the feature like milestones, that was a result of me trying lots of different things and learning from those things. Scrapping them and eventually arriving at something that seems to work really well.

Milestones, which is growing like crazy every week, and I think that's a result of all the experimentation. If I put any processes in place that would slow down that experimentation, I

might never have gone to this feature. Maybe it would've taken three years to get there. If I basically had a small team of engineers and I was telling them, “Hey, build this thing in the sort of a waterfall way,” and they just like went and built it. How would they feel if I just scrapped their thing the next week because I thought it wasn't good enough or I was constantly telling them mid-process, “Hey, I've learned of this new thing, this new principle, this new idea. We should completely change the plan for this feature in build it in a different way.”

I think like Indie Hackers, the product is in such an early experimental phase that it helps to be the founder sort of – I hate the term visionary, because it so like pompous, but it's like essentially you have to have a vision for where the product is going to go, and it's hard. It's easy to hire someone who can code. It's hard to hire someone who can like code and have the product vision that you have who understands the market, who understands your users and customers and the history of all the things you've tried etc., etc. So I think it’ll be a while before it's most efficient for me to outsource that job completely.

[00:34:58] JM: You think you’ll just feel it in the numbers overtime and feel it in the engagement overtime? When do you really step on the gas and hire a bunch of people and start amping up those dimensions?

[00:35:07] CA: Exactly. Because I think Indie Hackers has been growing, but it hasn’t been growing like a rocketship. It's not like hockey stick growth, right? There’ve been lots of things we've tried that didn't grow at all. So with something like Milestones, for example, that's working super well. We just introduce groups. Groups are working super well. The podcast is growing. A lot of these things are pretty mapped out and now I'm just making incremental improvements and we’re polishing these things and taking them in like the obvious directions. That's when you step on the gas and you really just ramp it up.

But there are other things that aren't working quite as well, like our written content. We’re constantly experimenting with. Right now, I'm experimenting with getting rid of like a whole separate blog and just sort of rolling that content until the forums or blog posts just look normal, sort of native form posts. That’s a whole experiment. I'm not really ready to step on the gas and hire people to make that go faster, because I don’t even know if it’s going to work.

If you look at bigger companies, a lot of times that's how bigger companies experiment too. They don't put the entire company on a totally new initiative unless it's like Google+ or something. They'll have like a kind of a skunk works team of small people, small team who can move fast, experiment, figure it out, go through that sort of that phase where you don't know how it's going to work, because you just need to be able to iterate quickly. For that, you need a pretty small team.

[00:36:17] JM: We were talking about games earlier, and you play a lot of StarCraft. There's a meta-game StarCraft, right? The meta-game changes over time. I think this is happening with media where, I mean, the meta-game certainly changed since I started Software Engineering Daily. I think it's change in my favor, where more people are listening to podcasts. But its changing in other directions. There's more content than ever. There's more podcast content than ever. YouTube's gotten really, really, really good. There's show much good written content out there. Hacker News has really become prominent. What are the meta-game changes that you're identifying right now?

[00:36:56] CA: I think there are two ways to look at things, and one of them is sort of a top-down approach. You're looking at these well-defined categories, okay? Media companies, or communities. Then you look at the established players. What are they doing? What's going on? You end up with like names for certain things that like only 5 or 10 years ago didn't have a name, like a social feed is a thing, and there are mechanics to how a social feed works. But like 15 years ago, no one would have ever said to put those two words together, social feed. What does that even mean?

The other approach is sort of the bottom-up analyze things from first principles approach where you just try to figure out like the essential isolated facts of the world. What is true about human psychology? What is true about what makes an app work or not work? Then you try to construct something that works the best for what you're doing based on those principles.

I'm much more of that sort of person. I don't like to copy what others are doing. I don't like to pay too much attention to the meta-game. I want Indie Hackers to be the best thing that it is. If that means it doesn't fit in any particular category that has a name, if it's not a media company or a social network, then it's just not either of those things.

Some of the best websites that I respect and look up to have also kind of done the same thing. So there's one that I'm sure many of your readers will be familiar with called DEV, DEV.2., started by Ben Halpern. DEV is unlike any other website that I've ever seen. What is it? It is a blogging platform? Is it a forum? It kind of looks like a feed on the homepage, but you follow people as well. Who knows what it is, right? They're just doing what works for them and they're not really caring what anyone else does.

[00:38:22] JM: That's right.

[00:38:23] CA: Yeah. I like to do the same thing for Indie Hackers. What's kind of cool, like I'm in this position now, right? I can build from an existing user base where I've never been in that position before. I've started so many different things that never really got that many users or had that much traction, but whatever I build now, I'm pretty certain that like a few thousand people are going to at least try it out.

I'm going to get to that point like no matter what. So I can kind of just like look at the data and make data-driven decisions, whereas in the past, it was more like lick my finger, put it in the air and guess which way the wind is blowing and hope things go right. Now I can kind of like look at my own user base and use that to inform the decisions I make rather than looking around what others are doing and just sort of copying their features.

I'll give one example, which is that there a lot of maker communities nowadays. Two years ago, there were no maker communities really. Now, there are a ton of maker communities. So Product Hunt has one called Product Hunt Makers. There's one called Maker Log. There's one called Work in Progress, and like plenty of others that like you've never heard of. Every one of them has the same exact paradigm, where it's like people post their tasks. So you would go on there as Software Engineering Daily and you'll say, "Oh, today, I needed to wake up." That's the task." Today, I need to edit this podcasts episode. Prepare for this one."

It's just like these social feeds of like thousands of people posting their tasks, and like they're not bad. I like especially Maker Log. I'm a big fan of it. But at the same time, I'm looking at these

tasks and I'm like, "I don't really want to read somebody else's task list. Maybe one specific person, I want like the high- level tasks they're doing."

But I don't really want to read their task list. But how do like all of these products end up with the exact same feature that I don't particularly find compelling? In my opinion, it's probably not a coincidence. It's probably people looking at what other products are doing and say, "Oh, that's a cool idea. Let me import that to what I'm doing." I think that's kind of the nature of humanity. We're all copy machines. We're all intrinsically wired to copy what we see other people are doing. I'm wired the same way, so are you. It feels good to copy and do things other people are doing. It feels bad to sort of branch out on your own and take a risk.

But I think if you can sort of build up enough of that institutional base level first principle's knowledge, then you can feel more confident taking risks and designing and building features that no one else ever has just because they match what you are trying to do and fit that like a glove.

[00:40:28] JM: How are your own media consumption habits changing?

[00:40:30] CA: I'm pretty deliberate about how I consume media. For example, I'm going on more podcast recently. I would like to do that. When I go on a podcasts, I would like to be able to like refer to things that are happening in the world and talk about what other communities are doing. So I'll spend more time deliberately reading about what's going on. Versus when I'm in heads down development mode, I don't really care about what's going on in the world. It's all noise to me. I don't care if I miss the news cycle. I don't care what Trump's doing. I don't care what's gone in the rest of tech. I just need to get this feature out the door. I need to do what I'm working on.

So I don't know if there's any greater trend that's influencing my media consumption habits more so than like what's going on in my life and how I need to sort of reconfigure things. I guess podcasts have been pretty influential however. I think podcasting is pretty much the new blogging. I don't really read that many blog posts nowadays. If I see a blog post on Twitter, I might click it and file it away and some bookmark tool, but I'll never come back and read it.

Whereas podcasts, I'll put them on playlist and like I will guaranteed get to that at some point in the future.

Audiobooks are big for me for the same reason. I can easily listen to these things when I'm cooking, or walking to work, or doing pretty much anything. So I think a lot of my media consumption happens over audio now. Whereas 5, 10 years ago, it was almost all reading.

[00:41:44] JM: There are so many people bending over backwards trying to figure out what podcasting turns into next and if it's related to audio interfaces or – Sorry. Voice interfaces, or if there's something else that's going to change. Do you have any bold predictions there?

[00:42:01] CA: I have none. Sorry to disappoint you. I would love to. In general, I think it would be good for more people to make predictions about the future. Number one, I think it's kind of fun. Number two, I think it forces you to really think. I have kind of a yearly tradition of – It started with like a New Year's resolution, but it just ended up ballooning into this like yearly New Year's planning. So like maybe 10 years ago I started doing this.

My New Year's plan will be like, "All right. First, figure out what you think you're going to be up to five years from now." Then think about like what you're going to need to do this year, in the next year, in the next year to get to where you want to be five years from now. It's endlessly entertaining for myself to go back and read what I wrote five years in the past about what my life would be like today and what my friends lives would be like today and what the world would be like today. Because I would make all these predictions and like oftentimes they were wrong. But I think sometimes they're right, and sometimes I made good decisions in the past that help me get to where I am now. So I don't have any like bold predictions about where podcasting is going. I probably should. But I would like to make more predictions like that.

[00:43:01] JM: Do you make bold predications about anything these days, or you just actualize your predictions by testing that in Indie Hackers?

[00:43:07] CA: I mean, that's exactly what I'm trying to do. I'm trying to make my vision of what I want Indie Hackers at least to be a reality. So I want to like kind of shape the world. I would love

for a significantly larger number of people to start Internet businesses. I would love for them to spend more time getting help and talking to each other online.

Right now, it's a very isolated world. You might sit on your computer working on whatever Jeff Meyerson works on 24 hours a day. Well, maybe not that much. But six days a week, whatever your work schedule is, and never talk to another person about what you're working on. I would like in the future to be sort of the default method to be collaborative, to be sharing what you're working on. I'm trying to create basically new habits and founders and make it by its own nature a very social thing, because I think the worst thing that a founder can do is isolate themselves. That's the default.

If you connect to other people, if you share what you're working on, if you share what you know and your tips and your advice and you also receive those tips and advice from other people, you're just going to be more successful. You're going to have more fun doing it. You're going to get luckier, because you're going to have other people who can help. It's just better in pretty much every way, but that reality is not really true today, or at the very least, it's true and sort of a not equally distributed fashion where like the most social capable networking type people are really good at it, but like everyone else doesn't really have a platform that helps push them to do this thing.

So in a lot of ways, I think with Indie Hackers, I'm trying to create this reality. I'm trying to change the way that people behave. So it's, yeah, less about predicting the future and then more about doing that whole cliché thing of creating the future. It's hard. Who knows? The chances of success of the world being exactly like I want it to be are pretty low. But it's definitely fun trying.

[00:44:46] JM: I think it's kind of like in some ways like Peloton. Peloton, people go there to get a bike ride. But then people find out that the instructors are inspiring. I think a lot of people go to Indie Hackers to initially find better tactics for running an Internet business or just top of the funnel figuring out the basics of what Internet businesses are.

There's a lot of inspiration on that, and that's one of your focus is actually. You've really focused on the whole idea of being inspired. Something I like about the site is that it's almost agnostic of

the business. It's more about business as one particular creative avenue where one looks for inspiration and looks for ways to paint a better picture, and the picture that you're painting happens to be a business. But there's a very creative thread that runs through all of it.

[00:45:46] CA: Yeah. I think a business can be anything. We got to think of businesses as being like this very stuffy, boring traditional profession, but like for me a business is just a project that you're passionate about that's also self-sustaining and that it generates enough revenue to allow you to continue to do it. That's all a business is.

Any other staff is completely optional. If you want to spend your time – I don't know, graffitiing walls and that's what brings you passion. As long as you find a business model to attach to that so you can continue doing that without starving to death, that's a business. That's not what most people think about when they think about businesses. But I would like to think that going into the future will sort of relax this stuffy definition and you'll see all sorts of fun, cool, inspiring stuff that people are doing and calling a business.

I like you brought up the point about inspiration. It definitely wasn't my goal when I started Indie Hackers. I was not like really even keen on what it meant to inspire someone. I was little skeptical of any talk about inspiration, but like watching the website work, watching what happened when somebody would share their story with other people who wanted to do it and seeing the people actually were getting inspired and like making life-altering decisions because someone else shared a story was mind blowing to me.

So within the few months of launching Indie Hackers I was like, "Okay, what is this inspiration thing? How is this actually causing people to change their lives? What are the components of it?" So I try to work that into every podcast episode. I try to work that into features, like the Milestone's feature is meant to be inspiring. You get an email in your inbox every day of, "A different Indie Hacker accomplished something cool, that's exciting," and you could tell them congratulations and you can hear their story. That's inspiring. The interviews on the website are inspiring. It's all meant to basically deliver the message to people that like, "Hey, you can do this. There are people who are like you who are doing this, and it's worthwhile to do it."

[00:47:27] JM: What have you learned about running a business from your time playing StarCraft?

[00:47:33] CA: I play a lot of StarCraft. God! I started playing StarCraft when I was like seventh grade, and I quit maybe four years ago. So what is that? Like age 12 to like 28 of playing competitive StarCraft? I'm a competitive person. I love playing StarCraft. I don't know how much I learned about business when I'm playing StarCraft. I think playing StarCraft was in some level a reflection of my work ethic, and then I practice so much, or it could have been the genesis of work ethic. Maybe I became a hard worker because I practiced StarCraft so much. Who knows? It's hard to say which one is true.

The other thing I think that I got from it is the idea of iteration and cycles. So in StarCraft you have like these very tight feedback loops. You play game. Within 10 the 30 minutes, you've lost that game or you've won that game. Instant feedback. Did you do well or did you do poorly? But even within a game, you're constantly building things and creating things and battling and things – You go into a battle thinking you're not going to win, and you lose, or you think you're going to – Something's going to happen, something else happens. You get a good feedback loop, and that helps you like adjust your strategy and basically reorient really quickly.

With the business, sometimes your feedback loops are super slow. You're like, "Oh, I have an idea for a company and it's going to do this and the product can look like this and it's going to have these features, and it's going to take me six months to get this out the door." It might take you like six months to get through one feedback loop just to figure out, "Oh, my idea was wrong, or the market was wrong, or it was the wrong time," etc., etc.

The advantage is that if you go through feedback loops faster, if you iterate through that cycle faster, you'd get better. So I was able to get like really good at StarCraft because I played so many games so quickly and constantly and every time I play the game I learned something. Whereas when I started my first company, I did like the traditional wrong approach of having super slow feedback loops.

They were taking me a long time to learn really concrete realistic information. It took me a while to realize of like, "You can start super small with a product or a company. You can do something

that's just tiny, that takes you a day to get out the door and learn a ton from that, and you're going to learn way more from that than you're going to learn by sitting around planning and writing up hypothetical strategies for how the world is going to work." Just get somebody out there and test it out and then iterate on that and go through quicker and quicker feedback loops.

So I think you look at why people get so good at games so quickly, why people are able to just improve at sports and games in general that have these superfast feedback loops and why people might start 10 businesses and never succeed. I think a lot of it has to do with iteration cycles. A lot of the people that I talk to on the podcast as well who've been super successful, they are people who do things like 12 startups in 12 months where they constantly are releasing a new thing. They're not happy just like working on one pet project and slogging it out for three years.

Of course, everyone's heard of like the few stories where that's been the case. But I feel like most of the people I talk to who've been successful, they worked on something small and iterated their way up to something big, or they just try lots of different things. Built up a lot of skills, a lot of knowledge as a result of that and then they were able to pick like a really good thing.

[00:50:32] JM: Why do Koreans dominate StarCraft?

[00:50:34] CA: Koreans dominate StarCraft because they have culture, in my opinion, that is more facilitating of being good at video games. They take it more seriously. If you look at when Korea got broadband, I think they still have the fastest Internet speeds in the world. I could be wrong about that. But for a long time, they did. They're way ahead of the U.S. in terms of Internet speeds.

When I was playing StarCraft as a kid, I would go home and sit on my computer alone and play StarCraft. When my Korean friends would talk about their friends playing StarCraft, they would go to an Internet café after school in Korea and play StarCraft with all of their friends. Of course, what's going to happen when you have like a social situation when people are all doing the same thing and sharing tips and secrets with each other? They're going to get way better at it. It's fun. It's kind of a parallel to what I want to do with Indie Hacker. I don't want founders to be

building these things in isolation. I want people to be talking to each other and doing it socially, because when that information changes hands, you just do way better.

I just read this book recently called The Secret of Our Success. Have you heard of it? It's a great book. It's basically this evolutionary biology book that argues that what makes people, what makes humans unique as species is nothing to do – I mean, a little to do with our smarts, but it's mostly to do with the fact that we experience what the author calls cultural evolution. So we pass information down from one generation to the next very efficiently and we're really good at doing that, but we're also good at doing a few other things and make this possible.

So we tinker a lot. We try lots of different things, and were really good at copying. This is kind of the essential mechanism that makes us humans. If a bunch of people go out in a tribe, I mentioned this earlier, and try something? You'll zone in on the person who's really good at something and you will copy them. It's gone to the point where we're evolved, were like we basically can't do anything. If I drop you off in the wild somewhere, like you'll be dead in the week. You definitely can't make it. There's no human with a big enough brain to just like survive in the wild anywhere unless they've trained a lot, which basically means that they copied what others have done and learned from others.

You can even run, really. If you go to – I forgot what this tribe is called, but like they basically run down gazelles and like hunt them, and it's like every other animals is just born knowing how to run. But if you want to run down a gazelle, what you need to do is look at like the elder person in your tribe and mimic his running technique, which he mimicked from like the best runner when he was a kid and so on and so on, back and forth like hundreds of generations, which involves like all sorts of weird stuff that you would even think about.

The fact that humans are like less susceptible to heat exhaustion than gazelles, so you need to run at the hottest time of day and you need to like run with an uneven gate and stop and start, right? You have to literally learn from others how to do something as natural as run, because we just are copy animals. That's how we learn.

So the upside of all of these being that if you put people in like a social situation where they can see who's really good and copy them, like they're going to do that. I think with StarCraft, that's exactly what plays out.

[SPONSOR MESSAGE]

[00:53:25] JM: As a programmer, you thinking object. With MongoDB, so does your database. MongoDB is the most popular document-based database built for modern application developers and the cloud era. Millions of developers use MongoDB to power the world's most innovative products and services; from cryptocurrency, to online gaming, IoT and more.

Try MongoDB today with Atlas, the global cloud database service that runs on AWS, Azure and Google Cloud. Configure, deploy and connect to your database in just a few minutes. Check it out at mongodb.com/atlas. That's mongodb.com/atlas.

Thank you to MongoDB be for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:54:21] JM: You know what's funny is I think one way that I've seen this cultural – Just tying together a few things we've talked about. I think that one way this cultural evolution does happen in games or in competitive environments are certainly in the still small world of startups and technology companies in Indie Hacker businesses is through this meta-game evolution. Even back when I played poker, like poker is such a small game. It's 52 cards, a hundred big blinds usually, and it's like the same game all the time. But the meta-game changed all the time.

What I think that says is that it takes us so long as humans to actually explore a space confidently and reach the most superior version of that game. I mean, you're looking at chess a lot these days. I'm guessing Magnus Carlson is like significantly better than anybody before him, right?

[00:55:22] CA: Yeah. He's arguably the best chess player who's ever lived, but he has access to supercomputers, and like the meta-game now for chess will be playing against the computer,

studying the lines what the computer generates, and it's pretty fast. I'm totally new to it, but there are certain moves in chess that are named after like 16th-century monks who originated that move and we can like look at them playing this move and we still call it by the same name as that monk. But there will probably be new moves and new patterns named after people who exists today, where many centuries went by where there wasn't that much innovation, because there's been like such a shift in the meta-game. If you really chess book from 50 years ago, it was different from 100 years ago. It was different. It just keeps changing.

[00:55:58] JM: This is what confuses me about competition. So we are also talking about competition a little bit earlier. If the game space of chess remains not yet fully discovered, that says so much about how much greenfield opportunity there is in the universe, right? Yet some people like to indulge in the competitive impulses of ourselves. You're talking about – I mean, we could talk about what competition actually means. But the person-to-person competition, there's something about person-to-person competition that is a fixed sum or a zero sum or a non-growing pie.

[00:56:42] CA: You win. Someone else has to lose.

[00:56:43] JM: Yeah. I don't know. What's the correct competitive framing? If you say you like competition, what does that mean?

[00:56:51] CA: That means I thrive on competition. That means that like at a base level I find it entertaining and enjoyable, but I also, I guess in a more productive way I'm pushed do more when I know that there is some sort of competitive aspect to it. For example, if I'm growing my podcast and basically like the darkness of space where it's just me by myself and my podcasts. If I go from a thousand downloads to 2,000 downloads, will I be happy about that? I have no idea, because I don't know if that's "good".

What I define is good or bad is entirely subjective. It's entirely relative to what numbers other people are getting. That's kind of what I take satisfaction. It's not necessarily beating other people or having them feel bad. It's more so knowing that like – I don't know. I just want some way to assess myself and it's really hard for me to do that without some reference point. That reference point is other people. I think that's the nature of competition.

[00:57:44] JM: Do you think starting a business is – In terms of pop-culture, do you think starting a business is becoming more normalized or is it becoming taboo?

[00:57:57] CA: Both at the same time. Actually, I think starting a business, like being in a nonbusiness owner and transitioning into the state of having a business is becoming more normalized and isn't really taboo. I think owning a business, especially making lots of money is becoming taboo. Just the political climate that exist is a little bit more classist today than I would say it was in like the United States that I grew up in, in the 90s.

Back then in the 90s, you could release a movie like Richie Rich un-ironically, right? “Oh, this kid is rich and it’s great and all of the poor kids want to play with this super rich kid,” and no one cares. No one bats an eye. Parents take their kids to see that movie. Today, that movie will probably not fly. He would not be the hero. He would be the de facto villain.

I think there's something to be said about like obviously politically we’re just more conscious of wealth disparity gradually gone that way over the last 20 years. A lot of that is inherently connected to business, because people who own businesses typically make a tremendous amount of money, because it's a tremendous source of leverage.

That being said, I think the Internet has this incredible democratizing force where now the barriers to building your own sort of business are much lower than they ever were, right? Most of the people who will, for example, be demonized by the press, like Jeff Bezos, or Mark Zuckerberg. These people are generally self-made billionaires. They didn’t have like billions of dollars passed down to them, like Jeff Bezos was, I believe, adopted and like made his fortune from nothing.

So it's kind of ironic that we live in an age where it's easier and more democratic for the average person to build something successful than ever was in the past, but it's also frowned upon. We’re in the age where it’s like, I don’t know, robber Barons passing down money to their kids. That was also the age in which business was looked upon as a more noble thing.

[00:59:46] JM: How possible is it to build an Indie Hacker businesses and non-software engineer these days?

[00:59:53] CA: Increasingly possible. So I interviewed Ben Tossell. He runs Makerpad. He has an Indie Hacker business. It's very meta. The entire business is devoted to teaching people who don't know how to code how to build some sort of application by using Zapier, or Airtable, or Webflow, or any of these tools that are helping non-developers like hook things up and connect them to different services on the Internet.

He himself doesn't know how to code and he used these different "new code tools" to build his website and he's now using those to basically provide a living for himself.

[01:00:26] JM: I want to interview that guy.

[01:00:27] CA: You should interview Ben. He's great. I think this is the feature. At the end of the day, people tend to coalesce into similar needs. For example, my mom wanted to build a website and I was telling her she should just use WordPress or something similar, where you don't have to code. So many people who build blogs, if you're building anything that's a blog-like at all, there's ways that you can do that without really knowing a ton about code. You don't need to hire a developer to build it for you. I think like the number of these tools is only going to increase. We're going to take the different tools and the different processes that we as developers tend to reinvent from scratch whenever we're building apps and we're going to have drag-and-drop tools to do a lot of that stuff.

I think if you want to veer too far off the beaten path and do something super custom, if want to build no completely unique product, you're always going to need code. But at the end of the day, there so many businesses, there are so many opportunities, so many niches that a founder can cater to on the Internet that don't require a custom product that I think we're only going see more and more people who are non-developers starting startups.

[01:01:26] JM: Yeah, it's like those – You used to hear about this – Well, I'm sure you probably still hear about these company, the companies that start a landing page, or a type form and then they gradually overtime build more and more functionality the type form or the landing page

feeds into and they gradually get there. Now you can just build a lot of functionality on that first day. I mean, I've been asking like people who work at bigger startups or bigger companies this question, just like we know how to use these tools as MVPs or we're learning how to use these tools as MVPs especially if you're a non-coder. Yeah, you get started with Webflow, or Airtable, or whatever.

If you're a big company, do you hire no code engineers? What do you –

[01:02:13] CA: I don't think so. I don't think you'll ever do that, because I think big companies generally just go really deep on things. If I think about my role as a founder, I'm wearing every hat. I'm doing the marketing, the sales. I'm doing the customer support. I'm writing the code, designing features, everything. It means I can't go that deep. I can only go so shallow, because if I go too deep in the area, I'm neglecting some other important part of the business.

When I look at a company, even looking at a company like Stripe from the inside, they're not really doing anything that I don't do as a solo founder. They're just doing it with like significantly more depth. They can hire specialists who can do like the thing that I'm doing way better than I'm doing it, with way more attention to detail. Google have an engineer who's optimizing like one tiny corner of YouTube, for example.

If you're going to go that deep, why have someone who doesn't know how to code, right? Why not just hire a specialist who can tweak all the specifics of that thing? I think a lot of the tools or people who don't how to code and people starting these no code businesses are pretty rough around the edges. They get the job done, but they're not like the best thing that you could build if you build it custom from scratch.

[01:03:18] JM: I wonderful if we get like – If this will turn into like code gen tools. Because if they turn to code gen tools, then it's all one in the same, right? You start with the Webflow thing and then you one click export it to a Rails app.

[01:03:34] CA: Then your Rails developer can take it from there.

[01:03:36] JM: Yeah. That seems like the way it will go.

[01:03:41] CA: That seems to make a lot of sense, although the end of the day, I wonder how much it's in any of these – There's so much psychology that goes around here where it's like, "What are the incentives of Webflow to do that?" If you're Webflow, you probably won't lock in.

[01:03:54] JM: It won't be Webflow. Webflow will be the Oracle of this space.

[01:03:57] CA: Yeah.

[01:03:58] JM: I mean, I say that with the deepest love for Webflow. They will own this space. I mean, for some period of time. Not own it, but –

[01:04:07] CA: Maybe they'll disrupt themselves.

[01:04:09] JM: Yeah, it could be.

[01:04:10] CA: You have to.

[01:04:12] JM: Last question. Give your boldest predictions – I know you don't like making perdictions these days, but your boldest prediction for how the Indie Hacker landscape will change in the next five years.

[01:04:24] CA: My boldest prediction is going to be pretty optimistic, because I'm hoping to be at least one part of this agent for change. But I think we're living – If you examine sort of the market of being in Indie Hacker, we're living in a time where technology is more available than it's ever been. It's cheaper than it's ever been. Information is also cheaper and more available than it's ever been. There's never been as much written about how to start your own business online. There's never been as much shared. The number of companies that are sharing literally their exact playbooks of how they've done everything and the amount of revenue they're making is staggering.

10 years ago, you'd probably find like only a small handful. So I think it's becoming easier to start Indie Hacker business than it's ever been. I also think there's lots of platforms built to

support Indie Hackers and then these bootstrap founders including Indie Hackers. There's also a lot of money of people, of investors who are looking to finance these things.

So five years from now, I think we're going to be seeing not only more Indie Hackers, but we're going to be seeing like more mainstreamification of this. I wouldn't be shocked if there are colleges offering degrees for like the Indie Hacker degree, right? Put on this hat and this hat and this hat. Learn these different skills to do that. I wouldn't be surprised if we start to see a shift and also like parents attitudes to their children's career paths.

Right now, everyone sort of assumes like you need to go to college to get a degree. That's sort of becoming less popular. I don't think is going to shift dramatically in five years. But like maybe 10, 15 years from now, like parents are more excited to see their kids go straight into the industry of starting startups as teenagers. Maybe five years now, everyone's like getting their kids on the Lambda school and high school, and who knows?

I think at the end of the day, the fact that it's becoming significantly more popular. I wouldn't be shocked if we just see it infiltrate the main streamers. Whereas right now, it's not. People are not talking about bootstrapping Internet businesses on TV sitcoms, really? It's kind of like the a joke that someone like started something in their dorm room and they're going to be the next Zuckerberg. But that's a whole different class of startup.

We're not seeing as much on the mainstream and people starting these smaller businesses that can sustain their lifestyle, but it's such a much more attainable goal for most people and so attractive to be able to start something that gives you your freedom and your ability to work with whoever you want wherever you want whenever you want. I'll be shocked if it doesn't infiltrate the mainstream. Infiltrate the mainstream at some point in the near future.

[01:06:45] JM: Not to open a can of worms, but you mentioned there's more and more investors looking to fund these kinds of business?

[01:06:50] CA: The number of emails I get from investors who are like, "Hey, you got a ton of companies. Would you introduce me to any standout companies?" It's insane. There's a lot of people with money would love to invest in high-grow startups or even just small startups

sometimes just for fun, because they like the aesthetic of doing it, but often times because they think it's going to be lucrative.

There are also these new, basically, accelerators that have propped up in the past few years including venture capitalists also who have like sort of an investment thesis of investing in smaller profit revenue- driven companies before sort of user growth traction-driven companies. So you've got NDVC, who's been doing this for a number of years. They're more of a traditional VC firm, but their thesis is around revenue-generating companies.

You've got Earnest Capital, who's started by Tyler Tringas and his partner who was a guest on the Indie Hackers Podcast. They're also investing in these smaller Indie Hacker businesses with different terms, and you would see venture capitalists investing. They're not like –

[01:07:44] JM: Dividends?

[01:07:45] CA: Yeah, dividends. I'm not sure the exact terms of all these different, because there's tiny seed as well. They have their own model. But essentially it's very founder-friendly. It's very slow-growth friendly. It's not I'm only going to get paid back if you exit your company for $200 million or a billion dollars. It's like if you just put along generating revenue at your current rate and grow five-X in five years. That's great for me as the investors. It's great for you as well. Pay me back, and we're all happy. I think we'll see a lot more of that coming up.

We'll see how it plays out. Who knows how these companies' first funds are going to turn out. But I suspect that they're going to find a lot of companies where founders could use the money. When I was building Indie Hackers, someone came up to me and said, "Hey, I'll give you 100 grand for you to support your lifestyle in San Francisco. But if this thing works out and you start generating revenue, you need to pay me back like up to five-X that over the next 10 years." I would say, "Great. That seems perfect."

[01:08:36] JM: I like those terms. Those terms are cool, where you just pay some multiple –

[01:08:40] CA: Yeah. I mean, there's like a cap.

[01:08:41] JM: Yeah. Pieter Levels has a pretty cynical take on this. I don't know if you saw that, but he compared these people who are investing at Indie Hacker businesses to old record labels. I was like, "I'm sure some of them are like that." But it seems like there's a diverse – Judging by what you said, there's the diversity. But I saw his tweet about that today and I was like, "Ha! I wonder what the capital –" I guess the dividend side of things.

[01:09:05] CA: I don't know if I would view it as old record labels. I mean, I didn't see his tweet. But the thing is when I think about a record label, I think of like – Because I think about almost everything as a business nowadays. I think of a musician as a business. But like they're a business where they're taking care of the product side of things, the music, and they're trying to outsource the marketing and the sales and like all the distribution and all the other parts of the business.

Of course, if you outsource that three or four quarters of your business to a record label, they're going to have like pretty onerous terms for you, because they're doing a lot of the work. If you were just a solo musician and you didn't figure how to get your song on the radio, then you probably wouldn't make very much money.

Of course, they're going to label deserves a lot of money. But I think for a founder, so when it goes into this thinking, "I'm going to wear all these different hats. I'm going to be responsible for the success of my business. The investors ain't doing it for me. They sort of enabling me, like they deserve some sort of reward for that, but they're not like literally the person in charge of my distribution model." So they're not as onerous as record labels, nor should they be.

[01:10:03] JM: Okay. Real last question. If somebody's listening this and they're going to go check out the Indie Hackers podcast after this, as they should. Certainly, the Pieter Levels episode is awesome. Any other episodes that that come to mind that they should immediately queue up?

[01:10:17] CA: I would say you generally can't go wrong by listening to the podcast in reverse order, because it's probably hopefully gotten better overtime. It's hard to pick a favorite, but I do like the episode I recorded with my good friend, Lynne Tye, who you know as well. We've been friends for 13, 14 years. It's the second most popular episode of the Indie Hackers Podcast

behind Pieter Levels' episode, which was out I think a year before Lynne's. So she might be catching up.

I think her episodes is great, because she is not someone you would expect to learn how to code. Yes, she learned how to code. She's not someone you would expect to start a business. She started a super impressive business. She has almost no expenses. She generates a crazy amount of revenue. Has a great lifestyle. Doesn't work that many hours. Just has a great story for how she got there.

I think it's really one of the most inspirational episodes, because it kind of shows that, yeah, business is hard. It's all a bit scary, but it's not that hard. I think a huge component of inspiration is seeing somebody that you can relate to who's not intimidating doing the thing that you want to do. I think Lynne is a very relatable great storyteller. Just great person to listen to on a podcast. So I recommend her episode. I think it's number 86.

[01:11:25] JM: Danielle Baskin too is a good one.

[01:11:27] CA: Oh, yeah. Danielle Baskin. She's awesome. She's also one of the top downloaded episodes and she's just like super quirky. She would've been –

[01:11:34] JM: Very quirky.

[01:11:35] CA: Yeah. She started – What? Like twenty something different businesses that she hasn't shut down any of them. She's just running them all in parallel. So her episode is a really good take on. I think the point I was making earlier where a business can look like anything, and anything that you're passionate about as long as you put a revenue model on that, it's a business.

[01:11:53] JM: Courtland, Thanks for coming on. Been a pleasure.

[01:11:54] CA: Jeff, thanks for having me.

[END OF INTERVIEW]

[01:12:05] JM: If you want to extract value from your data, it can be difficult especially for nontechnical, non-analyst users. As software builders, you have this unique opportunity to unlock the value of your data to users through your product or your service.

Jaspersoft offers embeddable reports, dashboards and data visualizations that developers love. Give your users intuitive access to data in the ideal place for them to take action within your application. To check out a sample application with embedded analytics, go to softwareengineeringdaily.com/jaspersoft. You can find out how easy it is to embed reporting and analytics into your application. Jaspersoft is great for admin dashboards or for helping your customers make data-driven decisions within your product, because it's not just your company that wants analytics. It's also your customers.

In an upcoming episode of Software Engineering Daily, we will talk to TIBCO about visualizing data inside apps based on modern frontend libraries like React, Angular, and VueJS. In the meantime, check out Jaspersoft for yourself at softwareengineering.com/jaspersoft.

Thanks to TIBCO for being a sponsor of Software Engineering Daily.

[END]