

**EPISODE 849****[INTRODUCTION]**

**[00:00:00] JM:** The software market changes every year. As individuals and enterprises become more willing to buy software, there are new markets for entrepreneurs to sell software into. Good software has high margins and high retention, so even a niche software business can prove very profitable.

As software spreads across the world, developing countries are showing a willingness to buy the same software tools as the developed countries, making the niche software businesses even more successful than expected. It's a great time to start a software business, but the fundamental challenges faced by an entrepreneur have not gone away. An entrepreneur must find the necessary capital, build an initial product, iterate on that product, market to their audience, sell to their customers and retain their users.

Hiten Shah is an entrepreneur and investor who has started several businesses both successful and unsuccessful. He's also the host of the Startup Chat Podcast, a show that I enjoy quite a bit about strategies and tactics for software businesses. Hiten joins the show to discuss his experience in the software industry, including lessons on financing, product development and the future of software products.

A few updates from the Software Engineering Daily universe. FindCollabs is a place to find collaborators and build projects. FindCollabs is the company a building, and we're having an online hackathon with \$2,500 in prizes. If you're working on a product or project, or you're are looking for programmers to build a project or start a company with, check out FindCollabs.

Also, I've been interviewing people from some of these projects on the FindCollabs podcast. So if you want to learn more about the community, you can hear that podcast. We also have a new Software Daily app for iOS, which includes all 1,000 of our old episodes, as well as related links, greatest hits and topics. You can comment on episodes. You can have discussions with other members of the community, and you could become a paid subscriber for ad-free episodes by going to [softwareengineeringdaily.com/subscribe](https://softwareengineeringdaily.com/subscribe).

Thanks for listening, and let's get on with the episode.

[SPONSOR MESSAGE]

**[00:02:32] JM:** LogDNA allows you to collect logs from your entire Kubernetes cluster in a minute with two kubectl commands. Whether you're running 100 or 100,000 containers, you can effortlessly aggregate, and parse, and search, and monitor your logs across all nodes and pods in a centralized log management tool.

Each log is tagged with the pod name, and a container name, and a container ID, and a namespace, and a node. LogDNA is logging that helps with your Kubernetes clusters. There are dozens of other integrations with major language libraries, and AWS, and Heroku, and Fluentd and more.

Logging on Kubernetes can be difficult, but LogDNA simplifies the logging process of Kubernetes clusters. Give it a try today with a 14-day trial. There's no commitment. There's no credit card required. You can go to [softwareengineeringdaily.com/logdna](https://softwareengineeringdaily.com/logdna) to give it a shot and get a free t-shirt. That's [softwareengineeringdaily.com/logdna](https://softwareengineeringdaily.com/logdna).

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

[INTERVIEW]

**[00:03:54] JM:** Hiten Shah, welcome to the show.

**[00:03:55] HS:** Thanks for having me.

**[00:03:57] JM:** The current business environment for software companies is very different than it was 5 years ago or even 2 years ago. What are the most notable recent changes to the software business landscape?

**[00:04:12] HS:** That's the really good question. When I first started in 2003 to 2005 is when we started sort of roughly exploring software, Amazon Web Services didn't exist. So you're talking over 15 years ago now, and we had to basically have rack servers at co-located hosting, basically, companies, if you wanted to really do something meaningful. We don't think about that anymore. There's a bunch of APIs. They're provided by Amazon, they're provided by Google, they're provided by Microsoft and many others now even. But on the highest end, the most popular services are all built on Amazon Web Services just because of the amount of things you could do that you weren't able to do in the past.

So it's actually a really big deal. The cost of starting something has gone down to very close to zero, and it's because of how easy it is to just basically use a bunch of equivalent of APIs on Amazon Web Services, and you're basically already got your site up. That's not to mention things like Heroku and other sort of products that are in the market that have even productized some of those APIs. So it's even easier.

I think that that's really the biggest number one dramatic shift that happened, which is the infrastructure to start things, the technical side of it, is almost down zero. Then you accelerate to more recently and you have this movement called the no code or low code movement, where there's a bunch of tools, whether it's a Zapier, or a product called Airtable, or a number of others that have come out that make it so that people can actually get very close to building applications and software without having to write any code.

**[00:05:58] JM:** If we think about the economics of computing, we can see that there has been an increase in the supply side. Thanks to cloud computing. So the supply of compute has gotten a lot better. Then on the demand side, we have mobile computing. The smartphone was – I think it came out 2006. No. It came out in 2007 I think, or the iPhone came out 2007. AWS was 2006.

So you have growth on the supplies. You also have growth on the demand side. Thanks to the increase in usage from smartphones. Do you think this growth in supply and demand, is this an explanation for this continued run-up of the stock market, this seemingly secular increase in public markets?

**[00:06:48] HS:** Yeah, I think when it comes to that, there is more consumer brands that are out there that get in front of consumers. As a result of that, we're seeing such a tremendous growth in terms of the companies that are in people's minds. So when you start like extrapolating that out, those companies are set in people's minds and then when they go public, people want a piece of it. I mean, this is what was happening back in the day too. The stock market is nothing new in terms of how it works. There're just folks that are everyday folks that can now get some level of ownership in these companies that are supposed to be growing so that everyone can make some money. Yeah. I would say that that's the case. I mean, that the other way to look at it is like back in 2003, there was barely 600 million people online. 600 million. Now, there's 4.3 billion people online. That itself explains the phenomena, right?

**[00:07:43] JM:** Yeah. I'd agree with you. So, in your personal business life, you have a portfolio of businesses. Some of the businesses in your portfolio are ones that you've started. You also have a number of investments. Describe your mental framework for balancing your portfolio of different businesses and investments.

**[00:08:06] HS:** Sure. So, I never think of it as a portfolio, but I think that that's a good way to think about it. So, I have a business called Crazy Egg I started in 2005, and I launched back then, and it's self-funded. It's still going. I have a business that is called KISSmetrics that I raised a whole bunch of money for and was the CEO twice, and then wasn't twice as well. That business had a lot of challenges and ups and downs.

Then I've started a few other businesses since then. One of the most latest ones is called FYI, and it's interesting that you ask this question for me. I think there're a number of those and I've invested or advised about 120 different companies as well. I think you mentioned balance, and to me it's more about where do I want to put my energy and what do I have energy for.

So I really like putting my energy at building the businesses, building the product, learning about how to grow these different types of businesses even though – They're all software, but they have some different things going on here and there. Also building teams, and I have energy for those things. So I do them. I don't think I'd have energy to run a coffee shop. I don't think I'd have energy to do something that's a lot more offline, because it'd be, in theory, a different kind of business, a different kind of way that you operate. At least with my businesses, all of them are

remote. So there's really no office, and that's how they're designed as of today, and that's been really helpful. So all my businesses since 2003 have been remote distributed businesses where people are all over the U.S., if not all over the world. That's been really helpful in being able to balance these, as you say, because we're doing a lot of messages on Slack now. A lot of calls sometimes to basically coordinate things and make decisions, and then a lot of things that are documented and processes that are created so that we can make things more repeatable. So, for me, it's always about the energy I have for these things, which oftentimes just means do I have energy for those people that are working in those companies and looking to make a difference.

**[00:10:09] JM:** The most famous historical example of a blend of investment and business creation and business development is probably Berkshire Hathaway. But as time has gone on, we've seen alternative models to this. What do you want to call it? Portfolio, construction, or balance. But then there're the more recent models. You have things like Y Combinator. You have Alphabet. You have Amazon. Do you find an affinity for any of these particular models or am I overthinking it? Is this too broad an abstraction? Do you just think of this as just kind of Hiten going through his day?

**[00:10:51] HS:** I'm just going through my day. I think that's probably the most reasonable way to say it. I'm just going through my day. I've been taking things as they come and I've definitely been – Either you can call lucky or been able to create some amount of success through these things, and I'll continue to do so and look for what I have energy for.

I am sure, over time, maybe in the next 5 years, maybe in the next 10 years, I'll have a better thesis, or a better example of what I want to be doing or how I want to do it for myself. I've studied all these models, but I don't think I have an affinity for any single one. I'll just look to learn from everything that I can see and everything that I find, and I think that there are certain aspects that have not changed at all in the world since the days of Berkshire Hathaway from back in the day.

A lot of the human side of things has actually not changed. While on what type of business and how the product and the businesses are delivered to the customer, that's changed dramatically. So there're a number of these things that I think through. So, if I were to say what do I really

spend a lot more energy and time on, is I spend a lot more energy and time on things that don't change. The things that don't change, or specifically people inside of these companies and how they operate and how they are able to get work done But the ways that we actually get the work done constantly keeps changing.

**[00:12:09] JM:** Continuing with the idea of this kind of models or formula of thinking through things may not apply to you in this context also, but there are different models for building software businesses. Just like there are models for constructing an investment system. So there's a model to building software businesses where you start with a really small target market like Amazon with books and then you expand into adjacencies. Then there's also this model where you focus rigorously on solving one particular product category and you just go deeper and deeper into that category and refine it over time.

I would think of Crazy Egg more in this category, where you have a very specific problem of the heat map on a website and the analytics around that heat map. That problem has enough depth that you can just build on top of it and refine it over time. But give a preference for his, because it seems like with Crazy Egg, you're kind of dedicated to that specific focus. You don't necessarily need to spin up adjacencies or expand into adjacencies. Maybe that's a property of the bootstrapping of the company. Do you have a preference for the model of expanding into adjacencies versus just doubling down on specific category?

**[00:13:26] HS:** Yeah. We call it self-funded, not bootstrapped. But either way, it's the same thing. It's a really good question. I think expanding into adjacencies is actually necessary. Crazy egg, we haven't necessarily done the best job of that in terms of capturing that and doing that early enough and realizing it. So I think one of the biggest learnings about Crazy Egg for us has been we have like user recordings. We have some other features. We are just not known for those.

So, it was more of a timing thing where we waited too long to add some of those features while other folks in the market have been adding them. More importantly, other folks have been adding the core feature that we have. Even today, if you go head-to-head on our core feature, ours is going to be better. We've just had more time to work on it, right? But we've made it harder for the customer to buy in a few areas. We've added some friction in a few other areas

around our marketing and things like that, and that's led to a great business, but also it meant that we are compared to these newcomers that aren't doing it necessarily better, but we're able to add our future. So they've turned our sort of one sort of feature into just a checkbox across many different features. That kind of stuff happens in a market. The feature gets commoditized. What do you do?

So I think for us, I would actually say that like I'm not necessarily sitting looking at Crazy Egg and saying we did the right thing there by being slower to add things. I think we did the right thing by building a great product that's very focused on one thing. We should've started thinking through how do we expand that much earlier than we did. Maybe even like as long as like almost 10 years ago we should've thought of that, maybe even earlier than that. So we're a little late to that party, a little late to that game and in some ways playing catch up right now.

**[00:15:11] JM:** I've heard you discussed differences between the long-term strategy for self-funded businesses versus VC backed businesses. What do you think about business that utilize debt? Do you think debt is underutilized in starting businesses, software businesses?

**[00:15:31] HS:** I think if you can pay back that debt or have a reasonable interest rate on that debt, enables you to grow and be profitable, there is nothing wrong with debt. I do believe it is underutilized. I truly believe that. But one of the reasons it's underutilized is many businesses just don't have a really strong understanding of their unit economics to be able to take on the debt and not go into debt, so called, and make it a problem.

What that means is you have a much clearer understanding of your customer acquisition costs and how much money you make and how fast from that expenditure. I see that over and over again. I think that that's one of the reasons like debt is not popular yet, is the ability for a company to explain the unit economics and the business is not as fantastic and not as high as you think it would be or should be by now.

**[00:16:23] JM:** It also seems like there might be an education problem or perhaps a technology problem. If you think about – VC has been in the Silicon Valley and the software business water for a pretty long time. So people know that this exists. People understand what kinds of businesses make sense for VC. Then Stripe came out. Stripe made it much easier to start your

self-funded businesses, and then that kind of got into the water. Then the downstream impact of that was a bunch of people starting “Indie Hacker” kind of businesses.

I guess I haven't really seen the debt technology evolve. I could imagine like a platform where, just as you say, “Oh, I login in and then it asked me some questions that help me figure out my unit economics or my projected unit economics, and then it gives me a loan maybe.” I mean, does that – I've heard of Cabbage. Maybe there're some other things? I don't know. Is it a technology problem?

**[00:17:17] HS:** I don't know if it's a technology problem. I think that's a business operations problem. I just think people just don't get it. They're not there yet. I mean, we're still new in software, right? we're still new in this world. Even with like direct to consumer e-commerce and categories like that, we're still new. We don't exactly know how these businesses scale and how spending money makes more money. That's what I mean by unit economics, right? These businesses are just not set up so that they have a really clear grasp of. If I spend a dollar today, I'm getting get three dollars back in the next three months from that dollar, right? That's why debt can be detrimental if you don't know how that works. You're right. That's what I like – Even today, you can't go somewhere and plug in your numbers and understand this. You have to go do that for yourself. It's not someone else's job to do that for you.

[SPONSOR MESSAGE]

**[00:18:16] JM:** DigitalOcean is a simple, developer friendly cloud platform. DigitalOcean is optimized to make managing and scaling applications easy with an intuitive API, multiple storage options, integrated firewalls, load balancers and more. With predictable pricing and flexible configurations and world-class customer support, you'll get access to all the infrastructure services you need to grow. DigitalOcean is simple. If you don't need the complexity of the complex cloud providers, try out DigitalOcean with their simple interface and their great customer support, plus they've got 2,000+ tutorials to help you stay up-to-date with the latest open source software and languages and frameworks. You can get started on DigitalOcean for free at [do.co/sedaily](https://do.co/sedaily).



One thing that makes DigitalOcean special is they're really interested in long-term developer productivity, and I remember one particular example of this when I found a tutorial in DigitalOcean about how to get started on a different cloud provider. I thought that really stood for a sense of confidence, and an attention to just getting developers off the ground faster, and they've continued to do that with DigitalOcean today. All their services are easy to use and have simple interfaces.

Try it out at [do.co/sedaily](https://do.co/sedaily). That's the D-O.C-O/sedaily. You will get started for free with some free credits. Thanks to DigitalOcean for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

**[00:20:18] JM:** Speaking of newer trends in software, there is this explosion of the centralized IT management provider. So like going from – There's this trend of going from the centralized Google system of using Google Docs and Google Slides and Google Mail and everything else, to using a cornucopia of startups. You are I think early to capitalizing on this trend with the new company that you're building called FYI. Do you have any predictions for how the explosion of new SaaS tools for managing startups and managing I guess later stage startups that are becoming enterprises? How is that going to impact the market for software?

**[00:21:09] HS:** There's going to be a lot more tools that people have to choose from across every category. I mean, think about marketing. I don't know. There's 5,000, 6,000 tools and growing in the marketing category. There's a big landscape called marketing landscape super graphic that's out. It's been updated.

Just as there's now so many, like over 4 billion people on the internet, and we didn't start that way, right? There is now like – I don't know the number. Actually, the super graphic is 7,000 different tools. Back in 2011, it was 150. I mean, just think about that.

The rate of growth, the proliferation of these tools across industries, verticals, and all of that is ridiculous. Now we have more options than ever. It's even worse than going to the grocery store and trying to buy things and commoditize categories. You don't see 7,000 options, do you?

**[00:22:08] JM:** Right.

**[00:22:09] HS:** So this is the thing. Then you look at sales. Now all of a sudden, sales tools. There's like – I don't know how many of them, but we're probably hitting the hundreds, if not thousands already in a few years. What about customer support? There are already a whole bunch of tools in that category? What about customer communications as a whole? What about email marketing? I mean, the list just keeps going on and on. How many CRMs are there out there?

The thing is, like yeah, you can say, "Okay. There's salesforce and the CRM category, but what else?" It's like, "Well, there's a million –" Not a million, but there's probably a thousand, maybe 10,000+ other CRMs that are out there that are targeting different verticals, different industries, and that's how it's going to be. That's the commoditization of software.

**[00:22:52] JM:** Now, I think of myself as pretty good at managing multiple technologies. But the proliferation of different tools has really challenged even me. I feel like I'm logging in with like 10 different email accounts across a thousand different technology tools. It's going into a – My costs are going into a variety of bank accounts. I feel overwhelmed by the management process of all of these different tools. Is that the kind of overwhelm that led to you starting this business, FYI, which is around managing documents from all of these different places?

**[00:23:34] HS:** Yeah. With FYI, if you go to our homepage, which is at usefyi.com, you'll see that what we say is find your documents in three clicks or less. We show you a whole bunch of the applications we integrate with. The thing we discovered is that companies and people are more and more using more than one tool to get the job done. So they used G Suite. They use Dropbox. They use Slack. They have files on their desktop, whether it's Mac or Windows. They have things in Evernote, because Evernote has been around for quite some time now. They might have things in Asana even. Things in Confluent. Things in Trello. Their documents are everywhere.

I think that like what's happening is that there are certain spaces like this one where it's really crowded and there's a lot of software out there, and the customer has choice. A lot of these products are free. So, anybody in a company can go pick up this product or any individual can

pick up these products and start using them and all of a sudden they go from having G Suite to Dropbox to Slack to playing around with Airtable. Now they've got stuff in Airtable. Then they can't remember or find where it is when they need it.

So that is the problem that we see. That is a problem that we saw. This is why we're very focused on solving this problem, and I don't think it's going away. I think it's getting worse, if you want to make it a negative. I don't know necessarily think from my perspective it's a negative. I think it's that these products exist. There's going to be more of them and they're going to enable you to do things that you can't do in other products.

So, of course, you're going to pick them up and try to use them and see what you can get out of it and see if you can make your processes, your workflow or the things you're doing more efficient. So I think that trend isn't just in this market. This trend exists across many different markets, not just in the document and productivity space.

**[00:25:23] JM:** So if I have a bunch of documents in Google Drive, I got a bunch of documents in Dropbox, I've got documents in Slack, all these different places. If I want to have those documents indexed on a regular basis, that problem has a lot of engineering depth, because there's a question of like, "How are you going to build that index in the first place? How frequently are you going to update the index?" When a new document gets added to Google Drive, for example, do you update the index immediately or do you kind of like wait 10 minutes and kind of have this rolling update system. Tell me about the engineering problems of FYI.

**[00:26:03] HS:** That's such a good question. So we have to integrate into – We have to integrate with all these tools. We're about to hit 20 different integrations. It's been about a year. So almost exactly a year and about a week since we publicly launched the product. So we've had to build 20 integrations. Those integrations need to be standardized to go into a specific backend on our end. Then we need to basically be able to display that for people in a really fast, efficient manner and deliver kind of on the promise of finding your documents in three clicks or less.

We also need to make it so that you can search. If you can't find your documents and three clicks or less with our interface, we let you search as well. So, the challenges are, I'm going to

say, unfortunately for us, countless. So this business is taking a while. It's not like a task management tool where you can build something in a few weeks to a month and launch it and then just keep going after it.

So we have a pretty robust backend. We have now about an engineering team of close to about 10 almost, and we're growing, and we need to support all these integrations. Now, think about it. If something changes with any of these APIs from these services that were integrating with, we have to account for it, otherwise our service breaks. Because our service is only as good as the things you integrate into it. There's no like non-integrated mode at the moment. I don't know if it makes sense to have a mode that's like you not integrating these things or not authenticating these things into our service.

So, we not only have to be really good at pulling in data from other systems. We have to be good at storing that data and then retrieving it at a fast enough speed and then we have to be great at the interface that we provide, because we're not a product that we're just selling it to the enterprise. We're a product that any individual can pick up and use and any business can pick up and use as well, and then we want to grow with these businesses and grow with these individuals as a useful product.

So, we've gotten ourselves into this business that requires a lot more depth than I actually initially thought whether it's because I was naïve or not. I think a lot of times what happens is you get started on something and then you realize the complexities and then you have to make a call on whether you continue or not. In our case, the problem is so big that it's worth continuing.

**[00:28:21] JM:** Yeah. That's what makes me curious about it, is like the first time I took a look at it I was like, "All my documents in one place. I am not sure I care about that or I want that." Then the more I thought about it, it's like this is something that has been fatiguing me for quite a long time. But then I returned to the site a number of times. I'm like, "Okay. Do I want to sign up for this thing? Make it a little bit through the funnel?" But then I'm like, "No. I am not sure I feel comfortable with a single point of failure in security." If I'm giving this thing authentication to Slack and Dropbox and Gmail, I don't know if I feel okay with that.

But at the same time, I think whether or not you convinced me or even if you convince a large populace of people, I think this problem is unique enough and I would say contemporary enough that whether or not like you end up solving document management. There is some problem within this overall set of problems that is a feasible business.

**[00:29:23] HS:** Yeah, that's the way we look at it, right? In terms of like the document, the hesitation and document security and all kind of stuff, we're not super concerned about that problem, because over time we'll make the people who are hesitant more and more comfortable. But we do have encryption. We don't actually store your documents, and we separate everyone's user data at all times. We even have a page that we don't call security. We call it protecting your documents, because that's what we're here to do. Otherwise, why would you trust us if we don't tell you that we're protecting your documents and tell you how we're doing it.

A lot of those things evolve over time where people get more and more comfortable, because the pain is so high, or other folks on the team are using it and things like that. So we haven't had that be a big concern by a lot of people. What we've seen is that people come to us and they want us to integrate with all the different tools that they use, right? That that's really where we have to like spend a lot more effort before we even worry about some of the other kind of issues around certain types of people and certain types of sort of security and auditing and things like that that we're going to want to do in the future.

**[00:30:32] JM:** Well, I like hearing about this because I can tell it's kind of a slightly intimidating problem. I guess you don't really know the engineering balance of it right now. I know the you have grinded on some difficult businesses in the past. So I know there was a hosting company that you put a million dollars into and you did not really get any customers.

**[00:30:53] HS:** When I launched it, yeah.

**[00:30:54] JM:** So you went down – And that's pretty amazing. So you went down that path and you said – I heard an interview with you where you're really honest about it. You said you had too much conviction about your idea, which is that happens. I've done that. I've done a thing where I put a bunch of money into a business and it was just – It's sort of like you build this

momentum, you build this conviction, you can end up going very, very far down a path without having any kind of traction. You can do these things with paid acquisition where it can really, really end up you burning yourself, convincing yourself you have a product.

When you reflect on that hosting business, were you completely wrong in your thesis for the business or did you execute poorly? What happened?

**[00:31:40] HS:** We didn't know any better. We didn't know the bounds of the problem at all. We just didn't know any better. I think this was one of the – We tried to build something really hard at the first shot. We had partners who we believe knew what they were doing, but we had no way to really evaluate whether they really did or not. We probably got emotionally invested in the fact that we thought it could be a big business. I think the emotional investment was probably the bigger issue on that one. There's a personal and emotional investment on just wanting to make it work, and I think that that was not an objective viewpoint.

**[00:32:12] JM:** Can you tell me more about the nature of that emotion? Do you think it was irrational exuberance or was there some other set of emotions that were provoking you to continue?

**[00:32:23] HS:** I think there is just a desire, and we wanted to pursue it. At the time, my cofounder and I, we're very relentless, and I think to a fault at that time. So, the emotion was literally just like we can make it work. We'll just make it work, and it was naïve of just making it work and I think we focused a lot on the fact that that market is big and it's worth doing versus what's our opportunity and being able to reevaluate consistently and constantly.

So at my current businesses, we're reevaluating things in a very healthy way almost all the time, but not in a way where we're like, "Oh! I'm sure of ourselves," but it's more like, "Hey, does this make sense? Does it still make sense?" I literally asked somebody on the team that, one of like two or three people on the team that every week, "Hey, does this make sense? Do we still have the conviction? Are the things we're doing correct? Are we heading in the right direction? Are our assumptions correct? Have we validated enough of them?" Those kind of like – Basically, all it is, is not asking the right questions about the business and whether the business is like on the right path or not.

**[00:33:35] JM:** How do you get people you're working with to tell you the emperor is not wearing any clothes? Because I mean you can ask them all kinds of questions. They may feel uncomfortable giving you honest answers.

**[00:33:51] HS:** I wouldn't necessarily ask like a person on the team. I specifically ask people who are in the leadership team. I think an individual on the team generally needs to be focused on what their one thing is that they need to be doing and getting them distracted from that is not a great idea. So this is more of like we have weekly leadership team meetings. We ask ourselves that in the meetings, or, personally, when we're thinking through something or working on like strategy for the business. For example, every few weeks, I have a strategy session with my cofounder. Today happens to be the day that I'm going to have that and there's an agenda, and the agenda looks like items that are potentially existential around risk and conviction and are we heading in the right direction or the things we should stop doing or the things we should continue doing. Things like that.

So, to me, I think this isn't about individuals on the team that have like specific goals whether it's like growth or an engineer on the team that's working on a feature or improving something like speed in the product. This is more about how you asked these questions with the right people on the team that can actually help you think through it and help just build a stronger level of self-awareness around kind of the business and where it's going and its viability.

**[00:35:06] JM:** One of the advantages of venture capital is that it can remove some of the – Well, I guess this can also be a disadvantage. But it can remove some of the emotional charge that comes along with deploying capital in a business that may be taking time to figure out the engineering problems. For example, in FYI, if you've got this set of engineering problems that are not completely solved, it can feel – If you're deploying your own capital. It can really get into kind of a gambling sort of situation. Do you have any good rules about how aggressively you deploy capital in your businesses when they're not making money yet?

**[00:35:51] HS:** Yeah, a lot of folks will do consulting or something like that in order to augment their income so that they can actually self-fund their business. I've seen folks do that and to great success too. So, I think there are alternatives. There some clear alternatives that are out

there and you can use them, and the biggest one is consulting, basically. That what I've seen the most common one if I were to give a prescriptive solution. If they do some kind of consulting based on the skillsets they have and they use that to self-fund the business.

**[00:36:20] JM:** So something that's happened to me is when I set up software business, sometimes they don't really like work. You start setting up and you build the product, and like you set up your Stripe account and you set up like G Suite sweetie and you set up some APIs and stuff, and then you kind of realize like maybe this isn't going to work. This isn't a good idea.

You get your business kind of halfway. You get in a half-baked state. Maybe there's a few un-paying users, but your business can add up in terms of cost. Maybe you get a bunch of hosting cost. You get a much SaaS cost, you got Gmail accounts, and you don't want to shut down these incomplete businesses, but they're usually costing you like 300 bucks a month or a thousand bucks a month.

I've seen this in a lot of people who startup numerous businesses on a regular basis. It becomes kind of a management problem. Do you suggest like winding down these kind of half-baked businesses aggressively or do you like keeping them around for optionality?

**[00:37:23] HS:** I don't prefer optimality. I prefer focusing as much as possible on the things that are working. If they're not working, make sure you have really good reason to continue working on them. I would aggressively wind things down if I were just to be really prescriptive about it. There's nothing worse than having something hanging in the back of your head.

**[00:37:40] JM:** So you mentioned this low code stuff earlier and these different APIs, these different platforms that people can use to build companies these days. You got Airtable and Zapier and so on. Are these tools good enough to build fully fledged, technologically sophisticated businesses even for people who are not really software engineers?

**[00:38:05] HS:** Yeah. I mean, it's getting easier than ever to like hire engineers and build something. So I think it's totally viable to build software even if you're not an engineer. I'm not an engineer. I fake it, and I fake it because I love working with engineers and I love building product and thinking through problems. Yeah, it would be tough for me to say no to that.



**[00:38:26] JM:** To me, these tools look like a higher level of abstraction. It seems like even software engineers should be thinking more deeply about how to use Airtable or how to use these other high-level things. How would you give advice? I know you're not a software engineer, but for software engineers who are deeply embedded in these lower-level ways of writing code. They think that they need to write JavaScript applications. They think they need to deploy to AWS. How can they explore more broadly and get to these higher level tools?

**[00:39:00] HS:** Yeah. That's a really good question. I think like you should know what people are using. You should just know what people are using. You should understand Zapier. You should understand Webflow. You should understand Airtable. You should like play around with code a little bit.

I think it's more about awareness. That way, you know what's out there even if you decide that it's not for you or you're not into it. If it can make your life easier, why wouldn't you give it a shot? That's like the more straightforward way to say it. If it exists and there're folks that are talking about it, why not just take a look? It doesn't mean you have to get invested in it. It doesn't mean you have to use it. Why not take a look? Just be open and take a look. I don't see an issue for an engineer with it. I think it's more about how do you make sure that you're aware of what's going on? As simple as that, in the world. Just like you would with like a new language that you might have gotten excited about or your friend keeps talking about, same kind of thing. Think of it as a new language. Engineers love like looking at new languages, right?

**[00:40:01] JM:** You host the Startup Chat Podcast with Steli Efti.

**[00:40:06] HS:** Yes, I do.

**[00:40:06] JM:** Give me your condensed theory on podcasts. What makes the podcast format so good?

**[00:40:12] HS:** You're in people's ears and it's very convenient for them to listen to it privately, and you're in their ears, literally. You're talking to them basically in a podcast. They can do it at the gym. They can do it while in the bathroom if they want to. They can do it in the car.

Essentially, do podcast, basically. Listen to podcasts anywhere, right? It's kind of like – I think if we didn't have such a huge sort of proliferation of like audiobooks, podcasts would probably not be as prevalent. But there was even books on tapes before there were podcasts, books on tape before there was Audible.

My opinion is that it's just an intimate sort of – A little more intimate sort of method of essentially communicating and transferring knowledge and learning. So it's going to keep growing up. I'm not sure where the market goes, because back in the day, one of our failed businesses back in the day was actually a podcast advertising network. I think back in 2008. So that's like 11 years ago now. Oh man! It was going to work back then. The market wasn't big enough. I think the market is getting bigger. That's awesome. That's kind of a big deal.

**[00:41:19] JM:** It's definitely getting easier to build these podcast ad networks, but I mean I interact with them a lot and they still have a lot of trouble. They have a lot of trouble convincing enterprise customers and – I don't know. It's a weird – It's very hard to see how rapidly the industry is changing. What would you like to see change about the podcast industry? What do you think needs to change the most?

**[00:41:43] HS:** Yeah. I think monetization is still nascent, and it's all about like partnerships, equivalent of like brand partnerships and things like that. So there are some work to do around tooling and things like that that make that a little more seamless. ROI is a little bit hard to get for folks with podcasting today. So, there's a number of those kind of challenges that – I would say that if podcasting, two things, one is the ability for people to make money from podcasts. That's a big one, and I think that we're really nascent there. It doesn't feel great to me.

Then in relation to that, the ability to get distribution for your podcast. I think it's really hard to put out a podcast and just make it work and expect it to get distribution. The channels don't exist, unless you already have existing channels. But then what happens is if you start a new podcast, you're spending a lot of energy looking to promote it and trying to figure out how to promote it, and that's a shame. I wish there was a better way to get distribution for podcasting and understand how a podcast grows and grow your podcast.

[SPONSOR MESSAGE]

**[00:42:55] JM:** When I'm building a new product, G2i is the company that I call on to help me find a developer who can build the first version of my product. G2i is a hiring platform run by engineers that matches you with React, React Native, GraphQL and mobile engineers who you can trust. Whether you are a new company building your first product, like me, or an established company that wants additional engineering help, G2i has the talent that you need to accomplish your goals.

Go to [softwareengineeringdaily.com/g2i](https://softwareengineeringdaily.com/g2i) to learn more about what G2i has to offer. We've also done several shows with the people who run G2i, Gabe Greenberg, and the rest of his team. These are engineers who know about the React ecosystem, about the mobile ecosystem, about GraphQL, React Native. They know their stuff and they run a great organization.

In my personal experience, G2i has linked me up with experienced engineers that can fit my budget, and the G2i staff are friendly and easy to work with. They know how product development works. They can help you find the perfect engineer for your stack, and you can go to [softwareengineeringdaily.com/g2i](https://softwareengineeringdaily.com/g2i) to learn more about G2i.

Thank you to G2i for being a great supporter of Software Engineering Daily both as listeners and also as people who have contributed code that have helped me out in my projects. So if you want to get some additional help for your engineering projects, go to [softwareengineeringdaily.com/g2i](https://softwareengineeringdaily.com/g2i).

[INTERVIEW CONTINUED]

**[00:44:47] JM:** In some of your businesses, your cofounder is Neil Patel, and Neil is complementary to your set of skills and building products and that he is quite good at sales and marketing of the products. I see a similar complementary relationship between you and Steli, because Steli has really focused a lot on funnels and sales and marketing. How does your relationship with Steli differ from the one that you have with Neil?

**[00:45:15] HS:** Yeah, Steli and I are not really business partners. We just do the podcasts and we do it for fun. So I think it's a lot different of a relationship.

**[00:45:26] JM:** What about philosophically?

**[00:45:27] HS:** Yeah. It's just different. It's like there's nothing there except we just sit down on a podcast and talk. With Neil as well as even my new cofounder, Marie, who I work with on FYI and another business of mine, Product Habits. The best way to say it is like when we figure out how to divide and conquer, things work out the best. They work out better than when we're trying to do both, trying to do the same thing. That was with Neil. The same thing holds true with Marie as well, which is like working on each other's – Basically, making sure each person is working with their strengths and really working on their weaknesses if they need to. But really focused on like what their best at and doing that for the business. I think that's a super important thing to do.

Even more recently, since like FYI is something we're spending a lot of time on, are going to continue doing it for the foreseeable future and keep iterating the product, Marie and I have basically split up responsibilities. While something like the Startup Chat, it's more like labor of love. We do it for fun. We do it because we enjoy talking to each other once or twice a week and recording these things and people seem to like it.

**[00:46:33] JM:** In that conversation, do you ever find that there is like a fundamental element of building business that you disagree with Steli on or is it just kind of a constant refinement and shared understanding of what goes on in businesses? Is there anything you could enumerate as actually disagreeing with Steli about?

**[00:46:56] HS:** Not particularly. I think we come to ideas together, and I don't think we really focus on the disagreement side of things. It's more of like making each other's ideas better. I think that would be the way that we treat the relationship we have and how we approach it.

**[00:47:13] JM:** What's the division of labor in FYI between you and Marie?

**[00:47:17] HS:** So, I focus more on growth sales and business development related things, and she focuses particularly on product content, if you want to call that marketing today. So that's the way we've sort of split up things more recently.

**[00:47:31] JM:** Interesting.

**[00:47:31] HS:** In the past we were both just working on whatever came at us.

**[00:47:35] JM:** How does that complementary relationship compare to what you have had with Neil?

**[00:47:41] HS:** I think with Neil, I was focused on product and engineering. He was focused on sales and marketing for the most part.

**[00:47:51] JM:** Neil is your brother-in-law. How do you avoid circumstances that – So like I've worked with my younger brother in the past and it's been extremely rewarding as well as extremely trying in certain circumstances and it's made me reflect a lot, like is it worth the risk? You certainly get a lot out of it, but there's a whole lot of risk too. How do you avoid circumstances that could lead to problematic fallout between a cofounder who is a family member?

**[00:48:22] HS:** So I'm married to his sister, and she's always been a great intermediary. So my short answer is she's always been a great intermediary and –

**[00:48:29] JM:** So you double down.

**[00:48:31] HS:** Yeah, exactly. That's one way. It's probably like one of the more interesting ways I would say, where we would – Early days, I think she was really helpful and helping us understand each other, and then at some point it just became easy or easier, because we just started understanding each other better.

**[00:48:48] JM:** Yeah. I mean, it really just raises the stakes. You could end up having – If you have a siblings, like I'm probably closer to my brothers than I am to my sister. I've never started a business with my sister, but I imagine if I did, it would probably bring us closer together or it could make us at our each other's throats. Just raises the stakes, but it could be a path to a better sibling bonding.

**[00:49:16] HS:** Yeah, exactly. I think it's more about how you keep things pretty clean in terms of the business as much as possible, and you know you have the family, the family relationship to fall back on. So I think there's a strong positive too as long as you're not holding baggage about each other from the past. Most of the stuff just boils down to how much emotional baggage do you have about the other person.

**[00:49:39] JM:** Both of your parents were anesthesiologists. Is that correct?

**[00:49:43] HS:** That's correct, yeah.

**[00:49:44] JM:** If you had to start a business in healthcare, what would it be?

**[00:49:47] HS:** I would probably target one of the more prevalent problems that exists in the world in terms of disease. Most likely something along the lines of like heart disease, for example. I'd go after like what are the problems that can impact the most amount of people and really go after that. I'm not sure which one yet, but like heart disease just comes to mind right off the top of the sort of top of the head.

**[00:50:08] JM:** Do you have it in the back of your head of like, "Dang! I grew up kind of seeing —" My dad is a doctor and I see him interface with the business world sometimes and it just seems like the way that the business world has developed in relation to doctors, it's a tragedy in many ways. But does it sit in the back of your head at all? A long-term wish to start something in healthcare or is it just not in your purview at all?

**[00:50:34] HS:** It's not. It's not something I think about a lot. I think my dad has been great at sort of convincing me since a young age to forge my own path and not worry about things like that. So I've been thankful about that. Maybe it even trends me away from healthcare medicine, but who knows what will happen in the future?

**[00:50:54] JM:** Your dad encouraged you to go into business because he said that as an anesthesiologist, he did not use his brain. I found that surprising here because I assume that there was in the work of an anesthesiologist, there must be some subjectivity. It can't just be

about memorizing facts and procedures. Do you take your dad literally when he said he did not use his brain as an anesthesiologist?

**[00:51:18] HS:** Yeah, the way he explained it is that most the things he did was muscle memory. That made a lot of sense to me.

**[00:51:24] JM:** Yeah. I mean, I guess learning technique over time, learning the rigors of technique. I guess that is different. Maybe it's not an absence of use of your brain, but it's like a different use of your brain. It's more just like forming muscle patterns.

**[00:51:38] HS:** Exactly. Yup. I mean, think about it. These are professions – This a profession where they actually starve you from sleep and you're expected to work half-awake on other human beings and save them and help them out, and that's just the way it is and it's fascinating. I think all you really end up getting is like really strong, great muscle memory. I mean, that's how you survive. That's how the job works.

**[00:52:06] JM:** In building businesses, I have made a lot of use of things like Upwork and Fiverr and hiring people through Toptal. There's an increase in the number of knowledge work gig economy platforms, and I'm unsure of how these compare in terms of kind of employee quality compared to full-time employees that have the equity incentives. I think there's certainly a conventional belief in Silicon Valley that you need to have these full-time employees. They need to have equity buy-in in order to deliver their best work. How much do you agree with that conventional thesis?

**[00:52:47] HS:** I don't believe in most convention in the first place, and I think to each their own. So like there is a factor of like start a company and raise a bunch of money. Bring on team members. Give them equity. A lot of these things are just the way things happen and the way things are. So they continue to be that way, because there's a lot of collective knowledge about how to set up those systems, and there are companies out there that sort of try to switch it up and some are successful at switching it up. But those tend to be outliers.

So there's usually in every kind of sort of business category, let's say, investment in startups that are sort of funded as an investment category or a business category. There's a certain way it

works. So you either figure out how it works and do it as close to that as possible so you look like everything else in the market, at least from a high-level, and can continue to get funding and be in that business or that sort of business model, or you do something completely different. We're seeing a lot of new different models being explored and tried out more often these days than not.

**[00:53:47] JM:** Give me a simple algorithm for pricing a piece of SaaS software.

**[00:53:52] HS:** I think one of the things people forget is the simple algorithm. I'm not sure if it's an algorithm, but is don't forget how your buyer buys today. What are they buying that's either exactly what you're selling them or close to it or related to it and make sure that you're aligned with how they're buying. I think that's one aspect that's more qualitative and fuzzy that not an algorithm necessarily that people forget, which is like don't try to get the buyer to buy in an unfamiliar way. Because that never really works

**[00:54:20] JM:** There is a phrase that in the software business, you should often try to make more money rather than figure out where to cut costs and improve margins in your business. Do you agree with that statement?

**[00:54:34] HS:** I think that answers both. It answers both. Find a way to improve your margins and find a way to make more money, and then you have two levers, not just one. I think depending on the business and the context of the business, what is probably easier than the other to do. I think a lot of people think getting more business, getting more customers, acquiring more customers, is easier to do than dealing with cutting costs or whatever, but they're probably wrong. There're probably a lot of inefficiencies. They just might not be willing to address those. It's just like this when the team member leaves your company, do you worry about what's going to happen before they leave? Because then you'd be driving yourself insane worrying about anybody and everybody in your company leaving, or do you just deal with it as it comes, if it needs to come, and try to prevent it obviously?

I think the preventive measures are smart. So when you think about it like that, it's like, "Well, there's probably a lot of savings that you can have just by thinking through the people that you have and thinking about are they doing the best job they possibly can. Do they really – Is a job



they're doing really that important? I don't mean to say that in any brutal way, like cut people or anything. But a lot of times the reason folks are not able to think about cutting costs is because that usually involves really thinking hard about the people, and we tend to – Whoever we are, we get some level of emotional attachment or involvement or like affinity towards people on our team and makes it harder and harder to let go of someone when they're not actually performing as well as they could.

**[00:55:58] JM:** You worked as an intern in high school. If the high school version of Hiten Shah were to walk into your office today and you had to manage that person, what would be your strategy for managing teenage Hiten Shah?

**[00:56:14] HS:** Two things. One, I would be very clear on why he needs to work on what he needs to work on and let him just figure it out himself. Number two, I would make sure that the feedback loop where I could give him feedback was very tight. So it was ideally daily or every couple of days.

**[00:56:31] JM:** Do you think he would be productive or is there just – Would it just be dead end, you'd have to let him go eventually?

**[00:56:37] HS:** I think it's if those two things were true and could be met, even though that might be a little bit of micromanagement in some cases around the feedback loop, although I don't think so, but some people might view it like that. Yeah, he could learn a lot really fast and be successful.

**[00:56:50] JM:** My sense is that in that internship you are quite bored and disenfranchised. Just as a closing question, do you think that we'll ever have different norms around working jobs versus starting businesses? I mean, it seems like it's getting quite easy to start businesses, but people still seem resistant to it. Do you think that norm will ever change? Will it invert?

**[00:57:16] HS:** Explain that a little bit more, where starting businesses will be easier?

**[00:57:22] JM:** Where starting businesses will become the norm rather than the exception to working kind of servile jobs.

**[00:57:30] HS:** There's a ton of this, like Maker Movement, even like freelance have been around a while. So, I'm not sure it will flip or people just have multiple roles in their lives where they're working. I think people will just have – It'll grow towards people having multiple roles in their lives where they're working. Not everyone, but like more and more people are obviously doing things outside of work to make money, outside of their core work, core job, their 9 to 5 so to speak.

**[00:57:51] JM:** Hiten Shah, thanks for coming on the show. It's been a real pleasure. I have listened to a lot of podcasts with you over the years and it's been nice getting to ask you some questions firsthand.

**[00:58:00] HS:** Oh, I really appreciate all the questions. It's always very thoughtful, and really looking forward to the reaching your audience and hopefully this has been great for them.

**[00:58:08] JM:** Awesome.

**[00:58:09] HS:** Thanks a lot.

[END OF INTERVIEW]

**[00:58:13] JM:** GoCD a continuous delivery tool created by ThoughtWorks. It's open source. It's free to use, and GoCD has all the features that you need for continuous delivery. You can model your deployment pipelines without installing any plugins. You can use the value stream map to visualize your end-to-end workflow, and if you use Kubernetes, GoCD is a natural fit to add continuous delivery to your cloud native project. With GoCD on Kubernetes, you define your build workflow. You let GoCD provision and scale your infrastructure on-the-fly, and GoCD agents use Kubernetes to scale as needed. Check out [gocd.org/sedaily](http://gocd.org/sedaily) and learn how you can get started.

GoCD was built with the learnings of the ThoughtWorks engineering team, and they have talked in such detail about building the product in previous episodes of Software Engineering Daily.

ThoughtWorks was very early to the continuous delivery trend and they know about continuous delivery as much as almost anybody in the industry.

It's great to always see continued progress on GoCD with new features like Kubernetes integrations so you know that you're investing in a continuous delivery tool that is built for the long-term. You can check it out yourself at [gocc.org/sedaily](http://gocc.org/sedaily).

[END]