EPISODE 321

[INTRODUCTION]

[0:00:00.4] JM: Developing product requires careful balance between engineering, sales, design, and customer service. The founding CTO of a company often needs to take on each of these responsibilities, because when the company only has a few people, there's nobody to delegate these different tasks to.

Cullen Zandstra is the CTO at FloQast, a SaaS tool for accounting close management. It isn't important what that means. We do describe what the product does in this episode, but the actual software product that we are discussing is less important than the general discussions that we have. Cullen gives some fantastic and detailed explanations for how to develop a software product from end-to-end.

In the past, Cullen worked at Myspace, and after we discussed his current company, FloQast, we discussed some of what he learned at Myspace and the product development at Myspace and why that social network proved to be less durable than Facebook. We also had some great discussions as to why Google+ was also less durable than Facebook despite having all the resources of Google behind it.

Also, I should mention that FloQast is hiring engineers in their Los Angeles office. FloQast is spelled F-L-O-Q-A-S-T, and Cullen asked that I mention that, and I think it was worth mentioning, because Cullen seems like a great product guy and FloQast seems a company where sky is the limit.

I also want to thank my friend Patrick Matheson for connecting Cullen and myself. Patrick is a good friend and a writer on Quora. If you want to read some great stuff about venture capital and other strange things, read Patrick Matheson's material on Quora.

I hope you like this episode.

[SPONSOR MESSAGE]

[0:02:03.4] JM: Life is too short to have a job that you don't enjoy. If you don't like your job, go to hired.com/sedaily. Hired makes finding a new job enjoyable, and Hired will connect you with a talent advocate that will walk you through the process of finding a better job. It's like a personal concierge for finding a job. Maybe you want more flexible hours, or more money, or remote work? Maybe you want to work at Facebook, or Uber, or Stripe, or some of the other top companies that are desperately looking for engineers on Hired? You deserve a job that you enjoy, because you're someone who spends their spare time listening to a software engineering podcast.

Clearly you're passionate about software, so it's definitely possible to find a job that you enjoy. Check out hired.com/sedaily to get a special offer for Software Engineering Daily listeners, a $1,000 signing bonus from Hired when you find that great job that gives you respect and salary that you deserve as a great engineer. I love Hired because it puts more power in the hands of engineers. Go to hired.com/sedaily to get advantage of that special offer. Thanks to Hired for being a continued longtime sponsor of Software Engineering Daily.

[INTERVIEW]

[0:03:28.4] JM: Cullen Zandstra is the CTO and cofounder of FloQast. Cullen, welcome to Software Engineering Daily.

[0:03:34.5] CZ: Thank you so much for having me.

[0:03:36.3] JM: Your company, FloQast, makes software for accountants to do close management. In this conversation, I would like us to focus on the process of developing a product idea and bringing that idea to a reality from the engineers' point of view. There are a lot of other podcasts that focus on the design point of view, or the end user's perspective, or the business, or the sales aspects of product development, and I do want to touch on those things.

The main focus of this conversation will be on the engineering and your role as CTO as you were developing FloQast, and perhaps the lessons that people can learn more broadly from this

particular cast study. Let's start form the top-down. FloQast is close management software. What does that mean?

[0:04:26.0] CZ: Yeah. The close in accounting terms is basically the work that has to get done in a given month for an accounting department. Really, what that close management software is really, at its heart, just workflow management software, that it's designed specifically for accounting departments. What that means in practice is we're pretty similar to typical work management software, like JIRA, or Basecamp, or Asana, except that we're designed specifically for accounts in the way that Envision is designed specifically for designers.

We'll kind of look at the very specific workflow of accounting departments and build something that's going to fit into that sort of keyhole really snugly and it kind of just accomplish the goals that the accountants are doing every month, but don't have a way to track, if that makes sense.

[0:05:23.1] JM: I love the envision analogy, both, because we had the Envision's CTO about a week ago, and also because Envision is this product where you look at the process of prototyping, and in some light, it sounds like a narrow problem where why would anybody buy software to do this design prototyping thing. There's such a narrow market, but it's actually a giant market, and FloQast seems similar. It's this very specific problem, but the closer you look at it, the more surface area you realize this problem takes up.

[0:06:01.7] CZ: That was something that I kind of learned in this process, is just — It's amazing that a SAS application like ours could be sold into the L.A. Lakers, who are our clients, and a gold mining company, who are our clients, and we sell into senior living facilities. Just all across the board, we have, basically, every protocol that you could think of kind of does accounting and subsequently needs accounting software, which is one of the things I find so incredible about SAS applications.

[0:06:34.9] JM: Close management is something that even applies to my business. I was looking at FloQast and I was thinking, "I have, maybe, 10 or 20 advertisers per quarter, different advertisers, and there's this protracted process of, first; you get a contract going, and then you get an invoice going, and then you have to clear that invoice, and that invoice gets cleared at some point while the contract is going on. You've got all these lose ends, and nobody wants to

deal with all these lose ends by just interacting with their inbox. I think that's what a lot of these narrow, or seemingly narrow SAS tools end up doing, is there is so much work that you do through your inbox and through Microsoft Word, or Microsoft Excel, and what these narrow SAS applications tend to do is they strip out some functionality and they make it like 10x easier. There are lots of accounting tools, why wasn't this particular problem solved before?

[0:07:42.5] CZ: I think that it sort of came on the cusp of the advent of the cloud and cloud accounting specifically. A big part of what we do is bring together disparate data sources. Now that people are storing all of their financial documents in places like box.com, or Dropbox, or Ignite, or whatever it is, and they also have cloud ERPs. All of their finances are now in the cloud and it created this incredible opportunity where a software like ours can kind of step in and bring together those different disparate data sources and put them kind of on one screen.

A big part of our business is sort of connecting the dots for people and pulling data from various different places across the internet and, to use an accounting term, reconciling it, which his basically just a comparison of that data and kind of alerting people to any mistakes or issues that they have sort of cross system. I think that just wasn't the thing when all of your data was housed in a specific spot, like in a shared drive.

Now that we're doing everything in the cloud, I think that there's just a tremendous opportunity for solutions like ours that are really geared towards looking at different systems and sort of aggregating data from different systems. We're kind of entering this new phase where it's great that we're on the cloud and we've got all these different systems doing all these different things, but how do we make sure that the data on all these various different systems is adding up and is what it should be? I think that this problem is not unique to accounting specifically, but it just so happens that accountings do a lot of different cloud computing at this point and we're kind of there to help you mange that process.

[0:09:33.4] JM: Did you understand from day one that this product was going to need heavy integration support from all the other tools that people use for accounting?

[0:09:45.2] CZ: Not at all. Not at all. In fact, when we first started the product, it was more geared to just a very specialized kind of task list and it kind of just goes to show that — It's

almost cliché to say that ideas are meaningless in startups, but, really, we have no idea that this was even a problem and it only came about once we started talking to people and you kind of get your first initial clients, they are like, "Yeah, we're kind of interested in what you're doing, but, hey, how about this other thing? This is the real problem for us. How do we take what you guys have already and augment it to include this kind of other issue?" Having those kind of early clients that are willing to work with you and willing to tell you their ideas. That's how all of these came about, and we didn't really know that it was a problem at all until we were out in the field and listening to people.

[0:10:44.5] JM: Let's talk about how you got to that point where you could have conversations with people which required you to actually build a product. How did you decide on that feature set of the first iteration of the product that you showed to potential customers?

[0:11:01.5] CZ: Yeah. Originally, kind of even before that, I had kind of quit my last job and was looking for a new startup, and therefore I was just getting pitched all the time by different people with ideas. I'm sure you know, as a developer, if you're looking to get into the startup world, you kind of have your pick of the litter when it comes to idea people.

While I was getting sort of pitched all these ideas, my now cofounder, Mike, pitched me this idea for accounting software, and he was an accountant previously, and he's like, "I have this specific problem where it was really difficult to track my tasks, or my specific workflow in the accounting department." At a point, you didn't really know if it was a real problem yet, but it seemed like a decent enough space to try to try something else.

We kind of just took that initial idea that he had from actually being an accountant and ran with it, built is a very small MVP. I think the MVP took me about, I don't know, a month of something to program with a very kind of crappy design that I put together in Photoshop, and I don't know Photoshop at all. Just that was enough to put into front of clients and say, "Hey, what do you think about this?"

I was surprised to learn just how receptive people are. If they do have a problem and the problem is big enough, they're happy to take a half an hour, 30-minute — An hour a week or

every month out of their day to kind of tell you about the problem and you can use that to get to the next level.

[0:12:45.1] JM: Talk more about that early feedback process with customers, because, at a certain point, you have to start taking their requirements and putting them into engineering things. You also have to figure out how to connect certain complaint that they may be having with engineering tasks to do. Can you explain that process in more detail and what you should expect to get out of your early customers and how you turn that into early product development?

[0:13:19.1] CZ: That's probably one of the more difficult things, because you'll — Especially when you're very first starting out, you'll hear all kinds of different ideas and you'll be pulled in different directions. At that point, you're just so desperate to sell some software that, basically, any idea that gets put in front of you that a client says, "I'll pay for that." You're immediately hot and bothered and you think, "Oh! This is it. I got to build this."

The important thing is to remember that it's important to get to market and get selling software as soon as possible, but at the same time, it's also just as important to really consider every move that you make, because your most limited resource is development time at that point.

The trick that we kind of used or, at least improvised, was just trying to get as much feedback as possible and cross-reference that feedback with even a handful of people that are interested in the product. You'll start to notice similarities the more you talk to people and the more similarities that you can kind of identify, then you'll know you really got something right. Once you hear the same thing two, or three, or four times from independent parties, it's like, "Hey! That's a problem," or like, "Oh! That could really be useful." Then, you can start to narrow down, and as much as possible, iterate, iterate, iterate, build the smallest possible version that you can and just get into people's hands. As long as you have a willing and able group of people that will test it out and give you their feedback, you can really trial and error your way into a decent product.

[SPONSOR MESSAGE]

[0:15:14.1] JM: Are you ready to build a stunning new website? With Wix.com, you can easily create a professional online presence for you and your clients. It's easy. Choose from hundreds of beautiful designer-made templates. Use the drag and drop editor to customize anything and everything. Add your text, images, videos and more.

Wix makes it easy to get your stunning website looking exactly the way that you want. Plus, your site is mobile optimized so you'll look amazing on any device. Whatever you need a website for, Wix has you covered. The possibilities are endless, so showcase your talents. Start that dev blog detailing your latest projects. Grow your business and network with Wix apps that are designed to work seamlessly with your site, or simply explore and share new ideas. You decide.

Over 100 million people choose Wix to create their website. What are you waiting for? Make yours happen today. It's easy and free. Just go to wix.com and create your stunning website today.

[INTERVIEW CONTINUED]

[0:16:34.7] JM: That's certainly true. I also feel like there might be a, perhaps, overly harsh narrative around premature optimization. What I mean by that is I think, for a longtime, people were mostly focused on this trial and error and just looking kind of around the corner at the near term optimizations that you can make. I feel like with increasingly higher level tools, more sophisticated tools, I'm thinking about platform as a service things, I'm thinking about open source libraries, third party services, like Twilio, that save you a lot of time. I guess, full disclosure, Twilio is a sponsor, but I always use them as an example.

[0:17:22.0] CZ: They're also our client.

[0:17:24.0] JM: Oh! They're also your client? Okay.

[0:17:25.6] CZ: Yeah.

[0:17:27.0] JM: It's also a service that just saves you a lot of time. The reason I'm mentioning these things is that because I feel like you get a lot of time savings out of these big platform building blocks that you can fit together, I feel like you can make — Even early on, you can make more ambitious roadmap decisions rather than just making short term decisions. I also feel like that can sometimes be valuable, because if you focus on the local maxima, you can end up with technical debt that is associated with that local maxima, rather than making longer term bets and asymptoting towards an architecture that will serve you better as you're going to scale. Do you agree that this is a tradeoff and that I'm touching on some relevant things, or do you think I'm blowing smoke here?

[0:18:15.2] CZ: I definitely think that there's truth to what you're saying that due to advancements both in technology and in services, that it's much more possible to do a lot with a little than it was even 2 years, 5 years, 10 years ago. I think that your margin of error when you're just starting out is just really low and if you kind of width on a product, it might be that the demoralizing factor kind of gets to you more than anything else that — You're there, let's say you're at an accelerator, you're just starting out, and you've got a runway in your bank account, or whatever, and you're kind of — Month over month, you're kind of just like watching that runway go down and you've got all these sort of pressure and anxiety about are you going to be able to sell some software?

At least for us, it was super important to just prove that we could get something working and get something that people were willing to actually kind of put their money where the mouth was an subscribe. Definitely, there's a lot that you can do now with services like Twilio. I think that it more speaks just to the kind of magnitude of what can be accomplished with a one, or two, person development team in startup environment, and I think that that kind of more speak to the progress of technology rather than the fact that you could width more often or something like that.

[0:19:51.1] JM: The more modest statements that I might be able to make about long term planning in terms of product development, might be around testing, or continuous delivery. You could put these things in place early on as long term bets or cultural practices that you want to instill even early on, even when you're building a piece of software for one customer, or zero

customers. How early did you start doing testing and continuous delivery, or do you do those things?

[0:20:25.7] CZ: Yeah, we certainly do them now. When we were still in the sort of ideation phase and it most important — Because you're always weighing what resources that you have most of. When you have more development resources and you actually have paying clients, then a software that works is your number one priority.

When you have a group of 5 or 10 people that are not paying for the software and are using it just because they have a problem and you're trying to solve that problem, then if it has bugs or if it doesn't work quite right, that's not as important. What's more important then is just that you're moving closer towards understanding what the problem actually is rather than having the perfect solution for the problem. I think on the outset of new tech venture to kind of put a lot of time and energy towards writing really maintainable solid software, a little bit putting the cart before the horse, always respecting jut what resources you have.

If you have the resources, great, do it. You'll really save a lot of time and effort later down the line with tech debt and what not, and we definitely experienced a good portion of — After we got off and running, we had this kind of MVC that I've built and I was fortunate enough to be able to hire my first couple of hires were gracious enough to take this kind of monstrosity that I had written and kind of shape it into a usable piece of code. You're just always weighing that, "Okay. What reasons do I have the littlest of, or the most, and how I can leverage what I have to kind of take the next step?"

[0:22:13.2] JM: We talked about integrations a little bit earlier. If you're writing accounting software, you have to integrate with Oracle, you have to integrate with Excel. How do you test those things? is it difficult to test around the black box of an externality that has close source code, like Oracle, or Excel?

[0:22:31.9] CZ: Totally. It absolutely is, and it becomes like an ongoing project, and we're still surprised. We're still surprised at something coming out of these ancient ERPs, like Microsoft Great Plains. It was literally written 20 years, 30 years ago, I don't even know. These ancient pieces of software that you just have no idea kind of how they work and you never stop sort of

iterating on that point, and every new client that we get, it's kind of a new journey into, "Okay. How does this work? What is your financials going to look like coming out of this system?" That's just part of this very unique business of just trying to integrate with these kind of ancient giants. Again, it's just like kind of trial and error. Every time we had a new client, it's a little bit of a mystery of the journey to figure out how — Not even just how the black box of Oracle worked, but how they're using the block box of Oracle. There can be many, many different setups within a single piece of software.

It has been said before, but the biggest thing is just having a great customer support team that will get to know the client well enough that they'll work with us as we kind of struggle with these kind of the black boxes that's mentioned.

[0:23:59.2] JM: This is something that may not be intuitive to engineers who are listening to this show, what is the communication process between the customer support team and the engineering team, because this is important, because this is where a lot of the feature development, or the bug fixes can come from, is closing that loop.

[0:24:22.5] CZ: For us, it was really important early on to get our customer service people really used to JIRA and the way JIRA works and being able to create tasks in JIRA and understand what iGMS is and how to properly document issues, and that's part of their training at FloQast, it's like, "Here's the development team. Here's how they sort of process new issues and tasks, or whatever it happens to be."

Those two teams work very much in tandem. I think, at most, most SAS companies specifically, because customer service is absolutely so important and it often times involve the developer hopping in and either seeing if there's an issue or at least understanding what the client is experiencing, or how we can improve or — Often times, the next idea will come from a conversation that a client had some misunderstanding about how the software works and that ends up leading to a whole new product, or something like that.

[0:25:27.7] JM: Do have any interesting practices around ticketing, and how ticketing facilitates communication the different teams?

[0:25:35.0] CZ: I don't know if it's necessarily interesting, or novel, but we definitely had a lot of work going to understanding how to classify tickets, like what gets priority, what doesn't get priority, what needs to be worked on by a developer, or what can be handled by, maybe, even a salesperson, or a customer support person to kind of try to navigate the situation if it's maybe a shortcoming of the software that we just can't get to.

It's an ongoing living process, but suffice to say, having solid ticketing and documentation is paramount. I can't tell you how many times we've gone back to old tickets to figure out how we've actually solved something for kind of what like what you mentioned with Oracle, or one of these other black boxes. We solved something six months ago, and now it's come up again. Having that sort of historical information has been absolutely crucial for us.

[0:26:35.1] JM: We've mentioned Envision earlier. In my conversation with the Envision CTO, we talked about how there can be between the design teams of a company and the engineering teams, you can have these silos that are somewhat similar to the development in operations silos that the whole devops thing is trying to rail against. How does the relationship between design and engineering work? Because I look at FloQast and it looks like a product where you have to get the design right, because you have a very specific customer in mind, you have a very specific workflow in mind, so that the relationship between design and engineering seems like it has to be pretty tight and neat there.

[0:27:22.0] CZ: Yeah, absolutely. We actually have our designers in with the development team in every respect there. They're at our daily standups. They're working directly with the developers there, at least, somewhat technically in clients. We want to have kind of information passed from one department to another as though through a pane of glass and have a constant communication between those two parties, just because, as you mentioned, the product design is so important to a product that is filling a very specific niche like ours, or like Envision.

I've worked at places before where I hardly even talked to product, or like the product would just come as a PRD that's already been sort of mapped out and thought out and finished in every respect and you would just get as a development spec and you just build it. You know what I mean? Certainly not so with the way I think startups like ours have to function. It's much more of a conversation and much more of a back and forth, and we've been blessed to have some

developers that were former accountants have kind of made the transition towards doing development now.

Having those people kind of looped in. In the building, I think that there's probably at least 10 or 15 accountants, and so we try to incorporate feedbacks from all of the different departments; design, engineering sales, support and setup. It kind of all goes in to what the product ends up being.

[0:28:56.2] JM: When you say you have 10 or 15 accountants in-house, are these people doing things like product management. How do you have enough — It's like a third of the workforce.

[0:29:05.8] CZ: Yeah. That's a great question. Another sort of fortunate about building accounting software is it seems that there are quite a lot of people that got into accounting and, now, realized that it's just not for them. Now, they're looking for something else.

A number of our sales people are former accountants. As I mentioned, our CEO is a former accountant. I think the majority of our setup and support people are accountants by design, because they kind of — In dealing with accounting departments, they have to be able to speak lingo and whatnot. We definitely geared our hiring towards favoring people with an accounting background, and we just been fortunate in the fact that, I don't know, for whatever reason, people are looking to make a transition out of accounting and get into, basically, anything else, but still leverage their knowledge of accounting. Yeah, I think we have, at least, 10 CPAs in the building, and so we've just — In that way, just been blessed.

[0:30:01.7] JM: That's pretty interesting. I've talked to a number of companies where they explain that the customer success people, or the onboarding people, are really well-trained professionals, and it almost sounds like you have to pay a good customer success person, perhaps, on par, or at least in the same neighborhood as somebody who is doing something like sales, or maybe even engineering?

[0:30:32.5] CZ: Absolutely. It's worth every single penny.

[0:30:35.9] JM: That's really counterintuitive for a lot of people listening, I'm sure, because they think, "Oh! Customer service, that's nothing." In fact, it's actually super important that a lot of these —

[0:30:45.5] CZ: It's everything.

[0:30:46.5] JM: It's everything. Yeah.

[0:30:47.8] CZ: It is everything. If you don't have a good customer service —

[0:30:51.0] JM: And you're not going to be able to pay — Sorry to interrupt you again, but you're not going to be able to pay an accountant, a trained accountant, like a call center salary.

[0:30:58.4] CZ: No. Absolutely not, and you wouldn't want to. Your customer service people are part and parcel with the success of the business. I just don't see how you could — I don't know. Maybe other people do it. I don't know, but I don't see how you could have a successful client relations without someone that is super knowledgeable about what you're going through and they can instruct you how to use a software. Then, on the backend, is able to kind of mine information from the clients about what we're doing right, what we're doing wrong, how we can improve. Knowledge customer success people, I think, is absolutely paramount.

[0:31:37.4] JM: It's also a good mote. I think good customer service is a really strong mote.

[0:31:41.0] CZ: Totally. Totally strong mote. Every time, we, on the development side, we kind of screw something up or cause some sort of issue where the site goes down, or something like that. Having a personalized relationship with these people, our customer setup people definitely taken the brunt of some angry emails, but just being able to — If you have somebody that you can reach out to and write that angry email and get a response back, apologize and trying to make amends however we can. I think that goes a really, really, really long way.

[0:32:14.4] JM: The two product expansion models I've seen for SAS companies that are solving a very tight vertical, such as FloQast, seem to be that you can either expand into adjacent markets, or you can ladder up and upsell your current customers into more products,

add-ons, better support, things like that. I'm not sure which of these strategies you intend to move towards, but can you explain how that strategic forecasting affects your current product development, or does it even affect current product development?

[0:32:56.0] CZ: Yeah, it absolutely does. We're definitely at the phase where ACVs are becoming really, really, really important.

[0:33:04.0] JM: Sorry. What? ACV?

[0:33:06.2] CZ: Yeah, average contract value. Basically, the value of each deal that you're sort of signing.

[0:33:11.2] JM: Okay.

[0:33:13.2] CZ: Basically, that number really determines a lot about how you can raise money. It says a lot about what the market is that you're going after. Whatever you can do to raise that average contract value, will really do kind of wonders for your company. As you mentioned, there's two ways to do that, and it's either upsells with new products that are in the same department, or kind of trying to expand, let's say, to different areas of finance, or maybe even something totally different.

For us, it's definitely something that we kind of grapple with and think about a lot. For us, the — At least to this point, we're kind of under the auspices that building upsells is probably the easiest way, or at least the low risk way to increase our ACVs. That's mostly because you can use your client base to shop ideas and shop new products with pretty low risk. Once you have a network effect, you can leverage that network in a really interesting way to kind of get a great sense of what is going to sell and what isn't before you even build anything.

We're at the point where there's a lot of low hanging fruit for us that we can just go to our client base and ask, "Hey, would you pay for this if we build it?" I think it's more difficult to try to land and expand into a different —

[0:34:48.8] JM: Food delivery.

[0:34:50.2] CZ: Yeah. Right. Right. Even like a different group of finance, because then you have to leverage your current connection and be like, "Hey, can you put me in touch with Bob and the CFO?" or whatever it is, as supposed to using the people that you've already developed a really solid relationship with to just sort of ask.

I think that there are certain limits to that though. You can't always upsell. There is that point where you can't just continually upsell into the current client base, and you'll have to expand to other people in the company, or whatever it happens to be.

[SPONSOR MESSAGE]

[0:35:35.1] JM: Good customer relationships define the success of your business. Zendesk helps you build better mobile apps and retain users. With Zendesk mobile SDKs, you can bring native in-app support to your app quickly and easily. If a user discovers a bug in your app, that user can view help content and start a conversation with your support team without leaving your app.

The conversations go into Zendesk and can automatically include information about the user's app information, device information, usage history, and more. Best of all, this is included with Zendesk for no extra charge. Use the out of the box iOS UI to get up and running quickly, or build your own UI and work with the SDK API providers. Keep your customers happy with Zendesk.

Software Engineering Daily listeners can use promo code "sedaily" for $177 off. Thanks to Zendesk for supporting Software Engineering Daily, and you can check out zendesk.com/ sedaily to support Software Engineering Daily and get $177 off your Zendesk.

[INTERVIEW CONTINUED]

[0:37:02.4] JM: I can also imagine land and expand happening organically where if so many people are using the product, some of the people that are using the product are like, "Hey, I'm having to bend over backwards to do what I want with this product, because it wasn't actually

built for that purpose." The reason that they're bending over backwards is because there is just nothing in the market for them, and so that kind of thing can signal, "Oh! Well, I guess we should build veterinary software, or something."

[0:37:31.4] CZ: Yeah. Yeah, and we definitely experienced that exact problem.

[0:37:34.7] JM: Really?

[0:37:35.6] CZ: Absolutely. We've had people kind of trying to shoehorn FloQast into a fairly different use case, and I think that that can be really enticing, but also something that you really, really, really, need to think about diverging. We're still a fairly small development team. There's about 12 of us now. We're not at the point where we can really focus on too many products at a time. Trying to diverge into or answering to request to shoehorn the software into something different. It goes back to all along the way, you really have to think about what resources you have and how to best leverage them.

[0:38:21.0] JM: Yeah, and your incentive structure — In terms of resources, if you're a company like Google and you have a giant profit flywheel, then, yeah, you can throw off $2 million on a crazy accounting software experiment for veterinarians. If you're a company like Uber, where your platform is based on subsidies and it only works if you build additional products on top of that platform, then you also have to take a lot of experiments.

If you're in the SAS — The sort of like SAS towards a vertical thing — I think your strategy of the vertical upsell as a potential growth strategy is a lot safer, it's a lot more sensible.

[0:39:04.6] CZ: Totally.

[0:39:05.1] JM: Yeah.

[0:39:05.5] CZ: Yeah. To your point, the other factor to weigh and all that, it's like, "What is your market size?" If you have a blue ocean and the market is wide open, then why bother trying to extend yourself and move into different compartments or move to different products. That just means that you need to scale up more on the sale side rather than the product side.

[0:39:31.7] JM: This is related to a topic I've been thinking about a lot lately, which is where you sometimes have to choose to be internally focused, or to be competitor focused. My perspective is that when you are in a green field market, it's okay to be totally internally focused, or maybe you want to call it customer focused. That's what Bezos always says, "Oh, we're just focused on the customer." We're not competitor focused. When you are in a market with entrenched players, and these entrenched players have entrenched workflows, you sometimes do want to be competitor-focused.

In certain areas of Amazon, you look at that light sale product that they released on AWS and it's just like they're trying to obviate DigitalOcean, or trying to disclone DigitalOcean, because it works. You do see Amazon do this sometimes when they take what looks like a competitor-focused approach because they see a competitor who's doing something right and they think they could do it better.

What are your thoughts on that? Do you have to focus on competing products, because you're in this very specific space, or do you feel like you're just customer-focused?

[0:40:43.2] CZ: I don't think that you honestly can be competitor-focused at all. I just don't think that it works.

[0:40:50.1] JM: Under any circumstance?

[0:40:51.7] CZ: Yeah, under any circumstance at all. I don't think that the model work, because there's so much that goes into building a product, and to look on the outside, that even if you understand the product well, to look on the outside at a product and then try to replicate that product. There's just so much nuance that goes into every small decision. Without having had developed that nuance, or an understanding of why the nuance is there, I just don't think that it works.

Every time that we've looked at one of our competitors and thought, "Hey, we should do that." We always came to the same conclusion of, "We don't really understand why they're doing that

in the most minute detail." They're the ones that have talked to the client and really understand the problem, and they'll probably understand the problem better than we do.

I feel like it's very difficult to emulate a product well, and I can't point to any good implementations where I've seen someone kind of just like rip off an idea and have it go super, super well. I just think that when you're solving a problem like that, to look at a competitor, maybe it's a good idea to just keep tabs on what your competitors are doing and kind of understanding of where the market is going, but without having the firm foundation of why. Why is the problem. I don't see possibilities to build a great product.

[0:42:24.8] JM: You've seen these Instagram Stories thin, where Instagram just copied the Snapchat feature.

[0:42:32.6] CZ: Yeah.

[0:42:33.5] JM: Yeah. I guess for people who don't know, Snapchat had this super popular feature — They still have it. A super popular feature called Stories. Instagram, which is a Facebook product, copied the Stories, and they did it better than Snapchat. This is a brand new feature — This is a feature that hadn't really existed in social networks before, and Facebook just ripped it off and they totally were open about it. They said, "We're taking this feature, because we think it's awesome."

You look at that and it's like the execution is incredible. I do wonder if they're making a deal with the devil when they're doing the feature copying, and it feels a little embrace, extend, extinguish. I don't know. Obviously, the execution is wonderful. I love having stories in Instagram. You might have seen the Snap S1, where you see the results of stories going into Instagram. It looks like it has cut down the Snap user base.

Do you think that Facebook is making a deal with the devil here, when it's sort of implicitly saying that it is okay to be competitor-focused?

[0:43:39.0] CZ: It's a great question, and I'm also kind of surprised that they were sort of so blatantly come out and say that they're basically recycling that idea. I'm surprised, because

people use Instagram and Snapchat for very different things, it seems strange to me that they — Even, now, Instagram has stories that people would use them for the same uses, or whatever, or same content. It wasn't uncommon for people to have both a Snapchat and Instagram, and I don't imagine that people are going to start using just one, just Instagram instead of Snapchat.

It surprised me, because I'm hard-pressed to believe that Facebook didn't really understand what the use case was behind Stories in Instagram and how people would use them as supposed to how people would use them in Snapchat. Clearly, they did their research. Clearly, they determined that there was a use case that people would be jazzed about it.

Again, I think that they're trying to capture something different, probably, than what Snapchat is, and it might work for them. I think if they were trying to clone Snapchat and clone all of the use cases of Snapchat, maybe they could pull it off, but I would say 99 times out of 100, it's just not going to work.

[0:45:08.7] JM: I think it will be an interesting case study to watch, because it has added something to Instagram. It has added a dynamic and real time feeling of Instagram and an emphasis on video in Instagram that did not exist before, and it feels like they have snatched a little bit of the Snapchat thunder, but there does seem to be — Maybe this is imagined, but it does seem to be a competitor-focused here that — I don't know. It's like this is what got — This is sort of what got Google into trouble with the Google+ stuff, whereas like the competitor focus would end it up just being such a massive waste of resources.

[0:45:51.8] CZ: Absolutely.

[0:45:52.2] JM: I don't know — Different case study.

[0:45:54.7] CZ: Why would you say that is? I have my theories about why it didn't work, but I'm curious what your thoughts are.

[0:46:01.4] JM: Why Google+ didn't work?

[0:46:03.1] CZ: Yeah.

[0:46:04.0] JM: I think it was an execution issue. I think they could have done it. It was just something — Something went wrong in the execution. Maybe how they tried to blanket all of their services with integrations, and Google has so many product lines. You have to do this massive alignment and orchestration in order to get everything right, and it sort of felt like they didn't decentralized it well enough. They could have — I don't know. It's hard to remember in retrospect exactly what it was like at the time, but I think I remember just using Google+. It was like — At the time, I actually — I remember, I posted a couple of things about this. I was like, "Well, there goes Facebook," and, boy, was I wrong.

[0:46:51.0] CZ: Yeah. I think that it came down. Again, they didn't really understand the problems that they were solving. I think that — My previous job was at @Myspace, during the decline —

[0:47:02.4] JM: I was going to ask you about that.

[0:47:04.3] CZ: Yeah. When I joined, it was kind of like the high watermark of popularity, and I was there for the sort of the mass decline and hemorrhage of users. I think that the reason that Facebook was able to come along and swoop all of the users, because Myspace had everyone. We really had everyone. Facebook was able to serve all of that, because they understood the problem.

I can say @Myspace that nobody really understood what Myspace was, or Myspace was really trying to do. Myspace was originally a platform for bands, and then it was like, "Oh! Now, it's a platform for anyone that can copy+paste some HTML CSS until I get a web forum. Then, finally, they said, "Strike while the iron was lukewarm," and go back to being a music oriented software.

Nobody really knew what Myspace was, and Facebook was able to come along and realized what the problem was, and realized kind of what people should be using, or what people wanted to use, and build that. Without that foundational level of what people are actually looking for, which I think it is only this sort of hard earned, just like grinding it out and talking to people.

Ultimately, products are about people. I feel like it's very difficult to look on the periphery of how something works and build it and make it work.

[0:48:36.0] JM: What is that insight that Facebook had that Myspace didn't?

[0:48:39.6] CZ: I think that they lived it and they breathed it. You know what I mean? He was there. He was living the college life, or whatever, and understood, from a fundamental level, how people wanted to interact on the internet and how that social aspect of our digital lives would come together. Whereas Myspace, we really didn't. We had no idea. We had no idea why people use Myspace. They just did, and we decided to capitalize on it, or that was my kind of feeling as lonely engineer at a massive company.

It didn't ever seem like they had a full grasp on what problem they're solving, or what they were really trying to do, or what the heart and soul of the company was. I think that that's evident in the way that it kind of oscillated between what they were doing or what audience they were really going for.

[0:49:33.8] JM: Yet, I've watched a lot of talks that Peter Thiel has given discussing Facebook. I think he also talks about this in *Zero To One,* where he says that his thesis is basically Facebook is the identity platform of the internet, is how you associate your personal identity. I think that's true. Going back to the Facebook versus Google question, that Google+ versus Facebook question, it is interesting to note that you have your inbox in Gmail and that, in some sense, feels like an identity.

It's tough for me to identify what about my Facebook identity is so much more identifying than Google is. I don't know.

[0:50:21.1] CZ: Yeah, that's a good question and one I probably don't have the answer to. It's hard to say.

[0:50:29.0] JM: Tell me more about Myspace, because this was — It's just like — I don't want to say too early, because Myspace is sold for a lot of money, but I feel like one of the things Facebook had in its back was, I guess, people were getting better at understanding how product

development should proceed. Maybe this was sort of getting to be the post-Job's era where Steve Job had sort of pioneered, "Here's a way to do product development." That had something to do with it. Maybe it's like post-Gates', or early Google product development. You really have this mojo of how products get developed that allowed Facebook to move at a faster pace, because they understood that. Myspace was really in the primordial soup of the internet.

[0:51:11.3] CZ: It really was. I think the people forget how far back Myspace was, and how JavaScript was like — It was like, "Whoa! You have JavaScript." Do you know what I mean? That was how old MySpace was.

As I mentioned, I don't think out of the entire lifespan of the company, anybody really understood what Myspace was, or what Myspace should be doing. Maybe it was just before web products had become so defined and so concise in their sort of delivery, that I think everybody kind of learned a lot from Myspace, and it was definitely a kind of integral part of the internet, and the antecedent for a lot of companies and products today.

I think that you're definitely right in that kind of — Or that brand of very precise product development, or even web products in general, just didn't didn't exist at the early onset. I don't think that the people at Myspace ever — Even the ones that originally created it knew what they were building, really, they're kind of just — As far as I understood, throwing something together and kind of seeing if it would stick, which I guess is a lot of what startups are made. Still, it just never really dawned — it struck me that they had a really solid plan for what the product is or where it was going.

[0:52:36.6] JM: Cullen Zandstra, I want to thank you for coming on Software Engineering Daily. It's been a great conversation.

[0:52:40.1] CZ: Thank you so much for having me.

[END OF INTERVIEW]

[0:52:45.9] JM: A few quick announcements before we go. Software Engineering Daily is conducting our annual listener's survey, which is available on softwareengineeringdaily.com.

You can click on the survey link. The survey really helps us understand our listeners and gives us data that we can show to advertisers that help get us better sponsorship deals.

Also, the Software Engineering Daily community has started working on Mineranker. This is an open source newsfeed platform. We are trying to democratize the idea of a newsfeed so that the only newsfeeds in town are not necessarily Twitter, or Facebook, or any other centralized newsfeed. We'd like to make it possible for anybody to make a newsfeed.

You can check out the Mineranker Project at mineranker.com. You can check out and implementation of Mineranker at softwaredaily.com. You can find links to all of these stuff at softwareengineeringdaily.com. There you can also find a link to join our Slack Group, to follow us on Meet Up for future meet ups, and other information.

Thanks again for listening.

[END]