Stanford eCorner

Creating a Buzz Around B2B Software [Entire Talk]

Christine Yen, *Honeycomb.io*

03-06-2019

URL: https://ecorner.stanford.edu/?post_type=video&p=63703

Honeycomb co-founder and CEO Christine Yen spent a decade as a software engineer before creating her own company. She describes how her deep domain knowledge and relationships with like-minded software developers propelled her startup's launch, and shares how she built an energetic human architecture around a highly technical B2B product.

Transcript

- Who you are defines how you build.. So Honeycomb is a tool for helping you understand your complex software systems.. What does that mean? What does that look like? Well, right now if you're software engineer in the industry.. Maybe some of you have experienced this at internships.. When you're trying to figure out what's happening in production.. You typically have two types of tools.. You have monitoring tools which give you graphs, which give you dashboards where you scroll through lots of things until you find a graph that spikes in an interesting way, and then you can look for other graphs that spike in an interesting way or you have login tools which deal with all the text output from your normal applications.. And you go hunting for that needle in the haystack that will tell you what went wrong.. Honeycomb's thesis is that these are two things that have evolved in a direction that made sense from the constraints that we had in the 90s.. When you grip, when you had counters, when we're like all right, well, CPU and memory are very, very expensive so we're gonna do everything very cheaply..

It is now 2019, it is terrible to be constrained by the things that people decided 20 plus years ago.. So with Honeycomb, we're saying you can have both.. You can have not only these graphs that we're used to but also all the fidelity, all the extra metadata.. All that useful information in your logs that told you what happened.. It can happen together.. We use the word events just describe things that are happening in our system and we talk about Honeycomb as letting you finally see your production system in hi-res, meaning you can see the big picture and then zoom in and get really close and understand why and what and how to fix what's going on with your software.. To give you a little bit of an idea of why Honeycomb, why now.. We're really seeing a few overwhelming trends in the software industry right now.. One, technology, the only constant is change so on and so forth.. There's a bunch of technology choices that are becoming popular that are changing how people are deploying software and maintaining software..

Containers, microservices, Kubernetes.. All of these pieces of technology mean that the average software system these days have more moving parts.. It's harder to keep track of which pieces are interacting with which other pieces and where the problem is coming from.. Along with that while technology is often the catalyst, we're also seeing changes in the process of engineering teams.. Agile has been going on for awhile but this like CICD are changing how people think about how they commit master.. How they push changes to production and finally people where we're literally seeing people are now defining themselves as a DevOps team because they're recognizing that the boundary between developer of I'm going to write some code and push the master when I'm done.. And operator, the people who have to maintain that software once it's in production.. They're seeing that line blur and they're looking for ways to bridge those two disciplines were previously thought to be completely different disciplines.. And with these changes, people are finding that their tools are currently outdated.. To give you an idea of what people are using Honeycomb for..

Our company has been around for three years.. We have a number of customers that have used us to change how they're shipping software.. Often people come in first for an instant response used case.. They're finding out that when they're experiencing downtime their current tools aren't giving them answers as to why.. They can tell them, they can tell the engineering team that something is wrong but they can't figure out the combination of factors that are actually causing an incident.. Allow them to resolve the incident.. Aside from what brings folks to us that's often what gets us in the door but more and more we're seeing people use Honeycomb not just to figure out why their code in production is not behaving as

expected in an incident context.. But more like why isn't it as fast as I expect? Why isn't my cut as efficient as I expect? Okay, well let's go look at what it's doing.. Let's go look at outliers and why my software might be behaving in a certain way.. And then we're seeing this cool used case, this cool flywheel of engineering team's folding Honeycomb into their development processes..

Using it to drive decisions about what code to write in the first place.. What normal is? What use case is or customer use cases to build for and really seeing this really cool virtuous cycle build in in the development processes in order to ship better code not just faster like CICD promises but also being able to ship with more confidence that your code will do what you expect it to do once it hits production.. To give you a very high level overview of where Honeycomb sits and what it actually does.. We adjust a bunch of data.. This is logs, agents in your code, whatever.. We adjust a bunch of data.. We have a custom column store, those on SSDs.. That column store lets us slice, dice, analyze, visualize.. It gives you insight into your data so that you can start to ask questions about, okay, well why the latency spike? Oh well, which endpoint did it spike for? Which customer did it spike for? Is there some strange? Is it being impacted by one of my MongoDB instances rather than the entire cluster? And this I had included a video which may or may not work of just what Honeycomb looks like and it seems like it is.. Great, ignore the audio but what this is showing is it's showing us a graph being able to slice and dice by customer ID in this case..

As a platform, you may have heterogenous workloads that you care about differentiating between.. We believe in tools that don't just tell you the answer but help point you in the right direction.. And what we've done here is the user had said oh, this artifact is weird.. Tell me what's special about it? And we've gone through and we have done some statistical analysis to say, hey, that user ID, there's one user that pops out in that highlighted section.. And again as you're understanding of what's happening in production and why it evolves.. You go in, click through, find a trace of a single execution.. Get down to this level of detail that really helps me understand what my code is doing and I can continue that cycle.. I can take what I've learned here, feed it back into this high level question of why is my code not behaving the way I expect.. This is a recycle deck.. Our customers love us and to give you an idea of how far we are, what we've been working on..

Honeycomb has been in existence since the very beginning of 2016.. First year, we really spent building out our prototype and establishing the market and this is something that lots of entrepreneurial classes will tell you not to do.. But what we really saw is when we looked at the market.. People wanted log tools or they wanted monitoring tools 'cause that's what they were used to looking for.. We started talking about hey, observability is this thing that isn't really captured by what your tools currently allow.. Observability is about asking new questions of your systems as well as the processes and culture that supports it.. 2017 and 2018 has been taking that prototype, these ideas, this excitement, really starting to land, turn our early adopters into customers.. Seeing traction, learning lots of lessons about GTM especially as a developer tool in a B2B market.. And right now we're in the stage where we're really trying to feel like we've really grown up out of our awkward teenager phase into our young adult phase.. We know what we're doing, we know what our value prop is, we know who we're talking to and we've just have to show that it's repeatable and go out there and kill it..

So that's where we are.. Thank you for listening to the context and now can-- - Yeah, thank you so much Christine.. - Great.. - Thanks for agreeing to do a little bit of interview session with the group.. - We're gonna leave this-- - Fantastic intro.. Before I get into some of the formal questions, you and I were having a little bit of a chat before you were talking today, and I was sharing with about where I started off on a technical track and ended up moving into a product management track but you and I actually talked a little bit about that.. You're heavy on the technical side.. Lots of students that I interact with one of the top questions that they ask is do I stay on a technical track? Do I go on a product track? Should I do both? When would I do both? But you've had a really interesting perspective on that and I thought that would be a great way to open up.. - Awesome, to give you an idea.. I had a CS degree..

I have been a software engineer.. I identify as a software engineer.. Been working for about 10 years.. I would say for the last two years what I've primarily been doing is not software engineering but all that software engineering experience has really helped me understand the problem domain and the folks that we're working with.. It let's me go to customers and really understand their problems, understand what their software development practices might be like because I've been a professional software engineer in enough different environments that I can start to understand what is normal and what is not.. It has really helped me build up when working with customers an understanding of the right phrases to pull on.. If you're doing customer development, you're talking to a customer, they're telling you about their problems.. There's certain phrases that I'm much better at.. I'd be like, oh, tell me more about that salesperson or someone who is less familiar with software might not be able to.. - Got it, very cool..

So one of the top reason why startups fail is because of lack of product market fit and that's the most difficult thing to figure out.. You think you've identified an unmet need in the marketplace and it turns out wasn't really a problem in the marketplace or there was other competitors in the space or you saw the need differently.. So last time I checked, I think you guys raised over $15 million.. You're generating revenue but tell me a little about where this idea came from and how you took it from this initial seed of an idea and got it to a company that's now generating revenue and has done really well in raising some capital.. - Awesome, Honeycomb started from myself and my co-founder Charity working together at company called Parse which got bought by Facebook.. And at Parse, we were dealing with tons of unmitigated chaos.. We were a platform that allowed level developers to basically include a black box and suddenly get all the benefits of having servers and

databases and everything.. It meant that it's great for our users.. They loved us.. We had to deal with their chaos and we had to deal with, oh gosh, the shared resources is being slammed..

Who it's being slammed by, why? Who else is it impacting and how can we put a Band-Aid on it so everyone else's service is undamaged.. And this questions of what's happening, why? Let's tease it apart and isolate the problems that I talked about in that demo.. That's exactly what Charity and I and the other folks in the Parse team were doing day-to-day.. We were bought by Facebook.. One of the things that happens when big companies buy little companies is they pat you on the head and they're like, oh, you're so cute.. Here are the big kid toys.. And with Facebook, most of them were garbage for our use case.. Great for Facebook use cases, not for ours but one of them stood out.. It was called Scuba and it let us really quickly slice, dice, understand our system in a way that changed how we built, how we maintained our software, how we built our software moving forward, and I think for us really the impetus behind this was thinking about going back out there.. Away from Facebook, larger software industry and not having this tool at our fingertips..

'Cause we had our battle scars from trying to solve these problems with traditional monitoring tool.. Traditional login tools and it was just so painful.. We're like this must exist.. - Yeah.. - I think the thing that becomes easy to discount when a startup idea comes out of that environment is how much easier it is to push adoption inside an organization versus outside.. Facebook has the luxury of saying, ah well new engineer.. Here are the tools you're going to use.. This is why, these are the trade offs and these are the only tools you get to choose from.. A little different actually in the real world but it's been a lot of fun being able to start off with that conviction of we know that this the right problem to solve, that this is the right approach because we've lived it and we ourselves are true believers.. - Yeah, so tell me a little bit about how the founders came together..

How you started thinking about roles and responsibility and that dynamic 'cause that's the second reason why startups fail.. Wrong people are involved or they're in the wrong position and that's another really hard part to get right.. - Yeah, so Charity and I had worked together for a few years at Parse before jumping into this together and we actually didn't spend a ton of time working directly.. She was the first in for hire and she ran the backend software engineering team.. I was a product engineer who built our analytics product and most of our interactions early on were like me shipping something and then her being Ops coming over and knocking on my door and being like, "Christine, this thing broke.. "Help me fix it." And so it's funny because that relationship that we had is almost the exact sort of relationship we're trying to break down now with something like Honeycomb.. But I think that we a lot of mutual respect for each other both working on this product that we loved, that we cared a lot about, and we're especially excited to work with each other on a tool like Honeycomb because we both felt like we brought dramatically different perspectives in an important way that would help us make sure that we shipped the right product in the end.. Ultimately Honeycomb is a tool for software engineers with operational responsibilities or operational sensibilities.. Call themselves owners.. People who are comfortable in code, comfortable with shipping code but ultimately want to be responsible for how that code continues to behave in production..

So we're excited to work with each other for those reasons.. We're bringing those different skill sets to the table.. I think when we started out, we were not the sort of people to be like oh, well five years down the road, we're gonna be a 200 person company and you should be CEO and I'll be CTO, and we'll have these giant organizations underneath us.. We're very pragmatic.. We're very like, okay, these are these roles that need filling right now.. One of us has to be CEO and none of it will really matter because for this first year, we just need to build stuff.. And so Charity reluctantly took on this burden of being CEO and I say burden 'cause that's how she felt.. We were engineers.. Going and developing relationships with investors and figuring out how to sell are not things that really were exciting to us at the time.. We wanted to build a product that will change how people like us would do their jobs..

And I think it was actually perfect in the beginning because Charity had spent so much time as an operator developing this public voice.. She did a lot of speaking before with her work through MongoDB and was very well practiced at going out there in the industry and saying, hey guys, there's gender neutral guys.. There's something different.. There's something, you can be doing your jobs differently.. There is this new future, I've seen it.. Guys it's so great over here.. Guys, gender neutral.. And especially when that first year was spent building up the market, building up the understanding of why observability mattered and why Honeycomb's approach was important.. I think it was great to have her more publicly out there.. Myself more focused on the product..

In the last, this spring we actually changed roles.. So I'm now CEO, she is CTO.. - Oh wow.. - And I think that it feels on one hand, feels like a big change, right? Oh, changing roles.. Like what does that mean? What does it mean? Our focuses have changed.. On the other hand as a startup goes from baby startup to three year old startup, the needs of each role also changes, right.. The type of person that you need to be to get something off the ground that sort of CEO that you'd need to be to get the industry and get folks excited about this new vision can be a very different person than that CEO who needs to get to that repeatable sales model.. And so, I think because we started off with this very strong basic trust between the two of us and understanding of each others strengths and skills we have been able to both do what is best for the company.. - Yeah, so just following to that.. So I'm really fascinated with that dynamic and how did conflict resolution and disagreements so that just happen naturally? Did that take some work and the part two of that question is and I've talked a lot about Generation Z and Millennials how incredibly self aware they are but that's an incredibly rare insight to have the self awareness to focus on what the needs of the company are and be able to literally adapt and take on a different role..

Where did that level of self-awareness come from and just the dynamic of how the magic of getting this chemistry right.. - Charity has produced exceptional.. - Yeah.. - And she had been a manager for many years at Parse before and so all credit for that self-awareness I think really goes to her.. - Yeah, very cool, that's awesome.. Okay, third reason why startups fail.. They run out of money and can't raise money.. So of the universe of early stage companies in the world.. The last stack that I looked at maybe 10% if they're lucky get a meeting with the institutional investor and of that maybe 10% of that actually gets a term sheet and gets funded so raising money at the level that you guys have done.. Walk me through a little bit how you overcome that challenge..

My guess is people weren't lining up trying to throw money at you guys and how you were able to get the company capitalized to do all the great things you're doing.. - It's funny you should describe it like that.. This is another point in favor of working at an existing company for awhile before jumping into a startup.. Because charity and I had done our time, worked in a number of different companies.. Because Charity had for the last two years that she spend at Parse really invested in doing the conference circuit and talking about MongoDB and becoming a domain expert in MongoDB reliability engineering.. She was more of a known quantity and especially when or at least startups also involve bringing customers and getting folks to pay attention to you that was very much a point in our favor when we were raising out first seed round.. And I actually remember one of my first conversations with one of our early investors.. He was a casual acquaintance.. And I was like, "Hey wanted catch up? "How is it going?" He's like, "Oh, I'm so glad you reached out.. "I heard Charity Majors left Facebook..

"Do you know what she's doing next?" And I was able to sit back and crack up 'cause like oh, well I'm glad you asked.. We're working together on a startup.. I hear you have money.. I think since then it only gets harder once they expect you to have more results to show every step of the way.. Seed funding I think is a really important institution but at that point they're betting on you, they're betting on your potential.. They're betting on your idea and how excited they are about the market and a lot of these difficult to quantify things.. And one thing that is certainly true and that we have been able to experience ourselves is okay, well, two rounds down the road, what do you have to show for it? - [Man] Yeah.. - And it's a very different conversation.. - Yeah, awesome, cool.. So getting the product right and figuring out product market is super challenging..

Getting the right people onboard in the right positions is really hard and getting the company capitalized.. The other part that also is incredibly difficult to figure out is your go to market strategy.. So I'm wondering who your customers are, how the market segments and how you've been able to reach your market? How you figured all of that out and were able to start establishing value and generating revenue? - This has been an incredible learning experience for us.. So again, a couple engineers who knew that it's not as simple as if you build it, they will come and certainly at Parse, we were coming out of this being like, okay, we need to charge money first or else we would have to sell to Facebook 'cause we don't have enough revenue coming in.. Not a bad way to go.. I think it has been fascinating.. So we started off thinking well of course we're going to, and we've always priced based on storage.. A lot of other customers in the space will do things like price based on seat, price based on host, how many hosts you're currently monitoring.. And for various reasons we felt like those were how, a, pricing models put in place by business people not by engineers and b, a relic of how again systems work in the past and a reflection of how infrastructure looked in the past.. So we've always priced based on storage and we started out with this thesis that people who have sufficiently complex infrastructures have the budget and will have the need to pay for something, a tool that really helps them understand it..

And I think there has been a little bit of because of a lack of a really strong business voice early on.. We do a lot of exploration and we had a lot of early success with folks who had been round the block, tried a bunch of tools and got it.. Who were like hey, I hear you.. You're saying Honeycomb is taking the right approach.. I believe in this.. I'm gonna place a bet on you.. Almost like investors do with seed stage companies and I think the thing that we discounted in trying to repeat that is the role of navigating an organizational structure, finding your way to that buyer from a practitioner who might absolutely get it but has no pull in their organization.. And so I would say that in 2018 especially the end of 2018, we've gotten a lot better at first finding a pressure release valve for those practitioners who are very excited but are never gonna be able to justify a large deal.. So okay, we've released a self-serve tier.. You can self-select into this..

You don't have to talk to a salesperson and you can just get up and going on in and on your way.. At the same time, we've gotten a lot better at understanding, first identifying who our buyer is, what our entry point is.. Who to talk to and second not framing the value of Honeycomb in terms of our technical capabilities but what it will enable their engineering team to do.. Very natural, we're engineers.. Hey, remember how you wanna try to do this thing and you couldn't and you're on call and your life sucks.. We have a solution.. That works really well for the person who is actually being woken up, not so much for the CTO or VP of engineering who's like, oh on call.. I remember when I was on call X years ago.. And so it has been an education and a revelation in figuring out how to speak to that person as well as how to assembly the right team of folks to help us speak to that person because Charity and my brains are wired precisely the wrong way constantly.. And so it's this constant unwiring that we have to do when speaking to buyers..

- Yeah, that takes a lot of discipline.. - Yeah.. - That's fantastic so as I look out into the audience, I see future entrepreneurs like yourself.. What lessons would you share with them about your journey? What have you really learned? That top three things that you think would be important for future entrepreneur.. - I'm going to give lessons to my past selves who probably would not have characterized herself as a future entrepreneur so I'm assuming all of you the fact that you're here, you are all

already past this point.. I'm gonna say these lessons anyway because I can.. As a technical person, it was very easy for me to be like oh, well marketing, soft skills, sales, soft skills.. I can figure those out and man, I wish I could go find my 18 year old self and smack myself upside the head and make myself go sit in the marketing class just to understand what playing pieces you have there, what levers you have.. What people in a different role care about and how that boundary should work, right? Because yes, I'm technical that's a great advantage to have but if you don't know why you are missing in these other roles.. Not only can you probably not do it yourself well right away..

It's also becomes much harder to find someone who can and building the right team.. Hiring, we've had a couple hiring missteps along the way because we didn't know what good looked like and what we were missing.. Another lesson I think I am, so you mentioned the company I started when I was 23 in 2011.. When I was 23, that startup was only ever myself and my co-founder.. I worked six days a week from around 11:00 a.m.. Until 4:00 a.m.. every day, and I know it was 4:00 a.m.. 'cause that's when the last bus left from Market back down to Lower 8th where I lived consistently.. Saturday nights, I would end around one or two and I'd do a shot and then go meet up with my friends at bar.. There is a level of sustainability..

Again, I'm hoping this is not news to any of you but there are things that you can do.. It doesn't mean that you should and probably means that you're not gonna be your best self when you do them.. Two of our first employees at Honeycomb were dads of toddlers so they had to go home at five to put their kids to bed.. - Stay and work at four in the morning.. - Sometimes they would come back online and work on their own schedule but I think with Honeycomb we were able to set the expectation from the beginning.. We are hiring you and in hiring you we are extending the trust that you are going to do your work sustainably and please yourself.. You are going not burn yourself out.. You're going to be responsible and do all the things that you should be doing but I don't wanna be having to tell you to go home at 2:00 a.m.. 'cause it's been the fourth night in a row that you've been at work.. Honestly, this is easy for us to say..

And Charity and myself again, probably have the most irregular, terrible schedules of everyone in the company but being able to set that expectation upfront that we do not expect it of the company I think have allowed us to hire a type of person that lots of companies are looking for ninjas and rockstars would love to hire but can't.. And I think the third thing actually then for future entrepreneurs is don't undervalue the culture of a company when starting.. Lots of other startup founders will say that their biggest problem is hiring.. With Honeycomb, granted we're pretty lean right now but we were lucky enough to say that with the exception of good marketing people who can market technical products, I believe the hardest role to hire for in industry.. We have been lucky enough to not really have had a problem with hiring because people come to us because they know that we respect our engineers.. We respect our employees.. There is an attitude of humility towards these other disciplines.. A salesperson isn't gonna join Honeycomb and be looked down on because we've gone through and learn all these lessons and been made humble by the rest of the industry and that frankly, I think our engineering team.. I think our entire company is half women and these sort of things that you establish very early on when it's two founders or two founders and two employees and that end plus one problem.. Establishing the right culture early in and saying that you care about the things that you care about early really sets the tone for hiring far down the road farther than you can see..

- That's awesome, fantastic.. In doing some research and looking in your background, talking a little bit last night and just in this conversation here today, you strike me as a polymath person.. Someone who is incredibly gifted and obviously, you're incredibly bright but can grasp and gain a lot of knowledge in a lot of different disciplines and that's pretty rare.. I worked with Elon Musk, he was able to do that but one of the things that I think is a testament to this ability and I wonder if you could tell the audience a little bit about.. Most folks can either be fantastic at B2B or super fantastic at B2C.. These are entirely different markets, entirely different needs.. The fact that you've been successful in both of those areas.. Tell me a little bit about what you learned through that and how that skill came about, and how you have this ability to be so successful in B2B and then also so successful in B2C.. - I would not characterize myself as successful in B2C.. Frankly, I think B2C is consumers are fickle and magical and who get B2C is they see the future in a way that I don't..

What draws me to B2B frankly is the conviction that I understand the problem and that's the power in being a developer and building developer tools.. I can immediately empathize.. I can smell the, I don't know if I can swear.. I can smell the pitfalls that we might fall into a little bit better.. And I think what I enjoy and this is certainly again my personality rather than anything is with B2B we have a little bit more of the time and luxury to really understand our customers, to really understand an account, understand what's special about their engineering team, about their practices, about the problems that face them, the problems that are unique to their business.. That's a lot of fun for me and really informs our product, informs how we think about reaching the next end customers like them.. And I think helps them feel like Honeycomb is really one of their partners rather than just a vendor or some tool that they used.. - Okay, cool.. What's critical obviously is recruiting and hiring really great talent.. You're only good as your talent..

And tell me a little bit about what do you value more.. Pure, raw engineering talent or domain expertise and how does your recruiting process work and how you reconcile those two items? - That's a great question.. I honestly would probably say neither one is the most important for us when hiring.. Raw engineering talent is important but not the most important thing.. I think and we actually have some post on our blog about our hiring process.. We would much rather understand how someone thinks and constructing technical interviews is a topic that I'm sure if you wanted to, you'd get a panel up here and take three hours discussing the merits of.. But there are so many dimensions of building software, shipping software as part of a team that whiteboard interview simply don't capture.. I will call out Stripe as another example of a really great company

that does their engineering interviews in a slightly different way where they accept that and we accept that software engineering is about reading code probably more than about writing code.. It's about pairing with someone to talk through a problem.. It's about understanding the architecture of the problem and especially for us, for product engineer, we want them to be thinking about what the experience is for the user..

We're a small startup, right now we're still at the stage where we are able to embrace people who aren't just give me spec and I'll execute on it.. We love people who can be given a technical problem but makes those trade offs and make those decisions based on what they think the best user experience is.. And so it's domain expertise is something that can be learned.. Again, as a developer tool, we have the additional leg up of lots of people come in naturally with domain expertise because they've had to deal with production incidents on the job.. And raw engineering talent is something that engineering ladders can address and ideally is already happening on the job.. But there's an element of curiosity and openness and acceptance that you have different skill set than I do but we can work together and pair and build something awesome.. That is I think what gets us much more excited about an engineering candidate.. - Very cool, tell me a little bit about your culture.. How you guys build it? Did it happen organically? Was it purposeful? What shaped it and how would you describe it? - I would say it is fairly purposeful.. So again, Charity and myself worked at Parse..

One of the things that I think became very clear to us was that Parse was very special to us because everyone who joined cared deeply about the problem domain and the way that we were addressing it.. There was a point when Parse was 20 people and everyone had built an app on Parse even the office manager.. That was cool and that was something that was really special and was something that had effortlessly come out of those founders.. So when Charity and I started Honeycomb, we were very aware of how the things that we did, the things that we signaled were important to us would shape the culture moving forward and how easily it is to let something spin away if you're not paying attention to it.. So again, model, we start off being every conscious of visibly and vocally rewarding the behaviors that we wanted to support while for example, working late, right? Charity and I worked late all the time but if one of our engineers who we knew had a kid was working late, we'd be like hey, you need to get home? There's no, thanks for working late.. I saw you tried really hard.. There's no thanks for working at 11:00 p.m.. on the day before Thanksgiving.. None of that.. We were very conscious of those signals that we would send because even though we knew those first two hires really well, well that next hire might pick up those cues and learn that well, I'm expected to be here at 11:00 p.m..

the day before Thanksgiving too.. We think that's terrible and I think that that extended to again work-life balance that extends to how we want engineers to work with each other in a very collaborative style rather than a leave me alone style.. And I think that we are constantly looking and seeing and course correcting and we've been very lucky to hire some managers who are similarly very conscious and invested in facilitating a culture of learning and curiosity to support the raw engineering talent that all of our engineers have.. - That's awesome.. All right, we have one final question and then we'll open up for some Q&A.. And I'm gonna steal Tina See-ligs, my favorite question.. Go back in time, you're having a conversation with your 20 year old self, what advice would you give that person? - Sleep more and have more fun.. (laughing) - Hallelujah.. - I worked really hard in college.. I don't regret it..

I'm sure I learned a lot but if my GPA was a couple point tens lower, would I have had some more fun? Probably, would it have made a difference to where I am today, probably not.. I think that in life, in work, it's constantly a game of prioritization, right.. If I have these 10 to do items, which one is the one that I can do that will have the most impacts so that I can go to sleep and not feel bad about the other nine.. That's something that I wish that I had perfected when I was in school.. Choose so that I could maximize for learning.. I could maximize for the biggest bang for my buck for things that would make my grades still be good.. But I look back and I'm like you know what, people talk about things like Thirsty Thursdays and I'm like, oh I always had lab or a giant p set or something due Friday mornings.. I had a very boring Thursday night always, man.. - That's awesome, fantastic.. Well look, I can tell you just from the short time that we spent together unless you've been snoozing through this thing, your energy, your positive attitude is so infectious, I literally and to have such a technical product..

I'd go work for you.. You made it so interesting and so cool and I hope all of you felt that way as well so we have a little bit of time.. Let's take some questions and we'll let Christine answer those for you.. - [Man] You mentioned early that you're getting better at finding pressure relief valve for the customer.. Did you have an epiphany when you were pitching customers the product early on, and find a better way to market or sell? - The question is do we have an epiphany that led to the pressure release valve for our customers.. I think when I describe a pressure release valve, I really think of it as a pressure release valve for our sales process and for building this GTM mention, GTM motion.. What we're finding and I would not describe it as an epiphany.. I would describe it as many, many protracted emotionally heightened conversations and try to figure out where to spend our time.. We basically saw is there was a ton of interest in Honeycomb but lots of it was from very vocal, very engaged two person startups.. And as much as we would have liked to build our business on two person startups, frankly developers of two persons startups tend to like to not spend money on developer tools..

And so, we basically had to find a way to let them use Honeycomb so that they could eventually it turns into marketing.. We think of it as a marketing expense.. Supporting folks who wanna us for free or for a very small amount of storage but who might be happy.. Might say good things about us to their friends working at bigger companies might turn into bigger deals.. And that was much more of how do we free up our sales people from not even having to think about these two persons

startups, and to enable the two persons startups to not have to talk to a salesperson which many engineers are still like, oh God, sales.. And it was again, maybe something that would have been obvious to a business person.. Someone who is more conversant in sales but something that we landed on and really embraced only after some time of struggle, yeah.. - [Man] When does the moment you decided to move out of benni-static and get into other things? I'm just curious about what are some of the learning experiences you had from there and into what you're doing today.. - Yeah, with benni-tastic, I think that and this is another lesson for you.. My co-founder at the time, I don't think that she really understood what she was getting into by committing to start a startup with me..

I think that she thought of it as something that would be a fun way to spend six months, and she ended up matriculating at grad school after we've gotten some angel investment.. And I was not confident in myself enough to be no, you're not allowed to.. You made this commitment to me, we're gonna do this.. And that's an example where I think if I can go back that would have played out very differently because ultimately, she matriculated at grad school about eight or nine months into that experience.. We tried to make it work and instead of, I ultimately walked away because I was like, you're not taking this seriously.. We're struggling to make this work.. We can't find some new idea that we're both excited about and I am emotionally and physically burned out and I wanna go just work with some of my friends on contract for awhile.. That was a lesson for me in the emotional investment that all of these endeavors take, and my rebound there was contracting with some friends with whom I cared about those friends but I didn't really care about the startup.. And it was nice to be able to just go and be an engineer and not have to feel emotionally tied up in that.. I needed to bounce back to that after the disappointment of having to walk away, yeah..

- [Woman] What are some tactics you use for efficient and effective collaboration? - Efficient and effective collaboration.. I feel like I want to list out a bunch of tools that we use but I feel like the key to that is having the culture and the expectation that that's how we're going to interact, not any one tool.. I'm going back and imagining some of my less collaborative co-workers in the past.. I'd pulled out a set of tools and been like, we're gonna do this.. It wouldn't have helped.. It's all about more like the ability to not take things personally when someone asks why and us both entering the conversation with an open and curious mindset rather than a set of tools which is an long answer, I apologize.. (audience applauding) (uptempo techno music)..