

One Knight in Product - E166 - Dani Grant

📅 Thu, May 11, 2023 1:38PM ⌚ 35:10

SUMMARY KEYWORDS

work, product market fit, jam, product, bugs, product manager, engineers, company, fast, people, talking, ship, founders, users, building, team, weeks, software, cloudflare, tonne

SPEAKERS

Dani Grant, Jason Knight



Jason Knight 00:00

Have you ever moved fast and broken things, or worse still moved slow and broken things? There were just so many things we could break. But wouldn't it be nice if we could just stop? Tonight, we might just find out how. Of course the people doing the fixing are the developers, but not everyone knows how to speak their language. If you're a non technical product manager or founder and want to get ahead in the game, why not check out Skiplevel. Skiplevel is an on demand training programme that helps professionals and teams become more technical in just five weeks, or without learning how to code, you can learn the knowledge and skills you need to better communicate with devs and become more competent in your day to day role with the skip level programme. So head over to <https://www.oneknightinproduct.com/skiplevel>, and make sure you use the referral code OKIP to support this podcast. You can check the show notes for more details. All right, so back to the episode and our whirlwind tour from product management to venture capital to kind of back to product management by starting a company of your own to fix a problem that was on your mind so much that you felt compelled to go back and fix it yourself. So it's time to strap in, put your crash helmet on. And let's start moving on One Knight in Product.



Jason Knight 01:15

So my guest tonight is Dani Grant. Dani is a former venture capitalist and product manager and co founder and CEO who says she hates pineapple on pizza. And as a former product intern at BuzzFeed, I'm sure be able to tell us why in a 10 part clickbait blog post. Dani wanted to be a fashion photographer when she was younger. She's now taking a keen eye for a shot and making sure the lighting is just right for her own startup as the co founder of Jam, a platform which he said has saved engineers over a million minutes of debugging. But you know what? Maybe those engineers could have just written it properly in the first place. Hi Dani, how are you tonight?



Dani Grant 01:46

Hey, Jason, thank you so much for having me.



Jason Knight 01:49

No problem. It is good to have you here. So first things first, you are the CEO and founder at jam. So what problem does jam solve? And specifically, how are you solving it?



Dani Grant 01:59

If you're a product manager, you probably report a lot of bugs to engineers. And you're right if you're most product managers, and you've probably felt the frustration of reporting a bug, an engineer picks up the ticket. And then they say I can't reproduce this and then they close the ticket. Jam makes sure that never happens. So with Jam, you have a browser extension that when you see a bug, you can just click on gym, it grabs an instant replay of the bug. So the last 30 seconds as a video, plus all the information a developer might need to debug. So for example, all of the console logs, all of the network requests, the specs of your device, your browser, your operating system, even your internet speed, and it packages it all in one link. And so that way engineers have all the information they need up front to debug, they can even see usually what the issue was right from that link. They don't have to even repro it on their own machine. And it saves you a bunch of time reporting the issue and it saves them a bunch of time fixing.



Jason Knight 02:55

Oh, there you go. So you're saying then that's primarily aimed at PMs, or just anyone that looks at software that has been developed. So like the QA teams, or anyone else that's just in the company, maybe they're doing a demo and something didn't work like is it basically aimed all of those people.



Dani Grant 03:09

When we started the company we were building for PMs, my co founder and I were product managers together at Cloudflare. That's where we discovered this was a tool that we really needed, but didn't exist. But today, lots of different roles use Jam. About 30% of our users are product managers, but also another 30% are QA and support. Another 30% are engineers and designers.



Jason Knight 03:29

Oh, there you go. You're expanding quickly. But the thing that you just mentioned about this 30 second replay thing. I mean, I think on a video that I saw demoing the software, you kind of refer to it as Twitch clips for testing, where you can again, record something that happened before you even knew that you want to record it. I do have to ask given the Twitch clips Comment, Like do you see that there's maybe a thriving industry for live QA streams in the future where people can start queuing software for Patreon money or anything like that?

D

Dani Grant 04:02

Not long ago, I would have told you no one wants to watch that I've reported a tonne of and it hasn't been particularly spicy. But recently a queue a competition approached us to sponsor and we're so excited to sponsor them. What happens with this competition is QA testers get together and compete to find the most quality bugs under a time pressure. And I can't wait to watch it. It's in June. And I bet it's actually going to be a lot of fun,



Jason Knight 04:30

It's something I'll put into my calendar straightaway. But I guess to be honest, there's no limit to what people are going to watch online these days. I was even thinking of starting to edit podcasts online as well just to see if anyone would throw a few, a few pounds in the jar for me. But you know, maybe we'll wait for the QA one and see how that goes first. But in all seriousness, are there privacy concerns with a tool like that like so? Obviously, if you're capturing stuff, effectively 30 seconds in the past, you're effectively having to click that all the time and then go back so you could be collecting quite a lot of data and that might make people feel a little bit, you know, kind of freaked out. So how do you avoid getting into that kind of trouble and making sure that people can have confidence in your tool.

D

Dani Grant 05:08

We're an ex early Cloudflare team. So we were all trained at a security company how to build products. And we don't want to take on more responsibility for users data than we have to. So we built that in mind, the way jam works is we take snapshots of the HTML shown on the page, we store it locally in the browser's local cache, and none of it reaches jam. And it's stored in the same place that like your passwords, your cookies, like your cache files are stored. So it's sort of normal use for the browser. And only when you say I want to share my last 30 seconds, do we compile it into a video and only when you want? When you preview it and say yes, create save it? Do we send it to Jim as just the last 30 seconds, and you can always delete.



Jason Knight 05:50

Well, there you go. You've got that covered already. But you've had a varied career. You've said already, you're a pm at Cloudflare. You're a product intern at BuzzFeed. You're an engineering intern at Harris Grooming, an analyst at Union Square Ventures, so a VC firm. So you've had a varied set of experiences, but still fundamentally working for other people's companies. So what was it that made you think, hey, it's time to strike out on my own and start my own firm. I mean, you said you had the problem. But what made you want to start a company to fix it?

D

Dani Grant 06:19

I love a big adventure and starting a company is is, is that in a nutshell, it, I find it very meaningful to go out and tried to solve something that I ran into and needed myself solver for

people and to get that positive feedback that we are helping their team be happier and more productive together. And also to build a team at jam where everyone is doing their best work together collaborating, like riffing off each other. And they feel like this is a meaningful part of their career and helping them make an impact to one of the really awesome things about our user base is, I get to spend a lot of time talking to our users. And behind each one of our users is a team of people who are trying to make some sort of impact somewhere in the world. And jam is the tool that they have picked to help them do that faster. And that I find that so meaningful and inspiring. And like, gosh, the things that they're building are amazing. Like, for example, the hospital system where my primary care doctor uses Jam And that whole hospital system runs on software, like I use their web apps, like book appointments, and message with my doctor. And for them to be able to catch and fix issues faster means that the hospital runs smoother, and patients get better care like that is, that's amazing. Or there's this really cool learning education software company building with Jam. They build special education software that can be more personalised for each student than a teacher who's trying to cater to a large classroom. And for them to be able to iterate on features and get them out the door faster to know that they are bug free. And working means that they can ship new ways of learning to kids faster. And that is like, I just I love that.



Jason Knight 08:01

So it sounds like you're changing the world, one application at a time, obviously, as you say, feels really meaningful and almost like a magnifying effect on top of the changes that you make, and even in your own product. So that does sound pretty cool. Do you use jam to test your own software as well? I guess you probably do, like ultimate dogfooding scenarios?



Dani Grant 08:21

Well, yes, it saves us a bunch of time. I love it. The one thing we can't do. So we have two classes of bugs. We have bugs that we can use Jam on because they happen in our web app. And then they have and then we have bugs within the Chrome extension itself that we cannot like Jam cannot Jam itself. And those bugs take us three times as long to fix because we have to do a back and forth, Versus if something happens in our in our UI, which is like the other core part of the product. It's instant. I mean, yes, we use it, and it's great.



Jason Knight 08:50

100%? Well, I'll probably give it a try after this as well. But why did you feel that you? I mean, I know you said that you were working, I think with another pm at Cloudflare. You kind of saw that this was a problem that you didn't have a solution for. But I mean, was that something that that was literally the origin story, like we just had this problem and we felt we had to create it because you didn't go and create that immediately from being a pm at Cloudflare. You went on to work at a venture capital firm for a bit, and then you came back to it. So like was that just nagging in the back of your head for a while or did something else kind of precipitate actually starting that company to do that?



Dani Grant 09:27

When Irtefa, my co founder and I were product managers together at CloudFlare, we were on this really special team. Our job as a team was to ship net new businesses, for the company. And to do that we had to move really fast and things that we shipped had to work out of the gate otherwise, it wasn't a real test if it was a valid business. And time and time again, the thing that slowed us down was all of the back and forth and miscommunication with engineers about fixing issues. And in the in person world. What we would do is we would go replicate the bug on Our own laptop then like, wander around the office till we found an engineer who could fix it. And then we would just hand them our laptop and they would debug on it. Because otherwise, how would you share the console logs and the network requests. But when the world went remote, we couldn't do that anymore. And I had moved on, I was working as a VC. But my co founder Irtefa, was still a product manager on that team. And we were just chatting. And he was explaining to me just the, like, the real struggle in March 2020, April 2020, of how they were trying to do through remote, and we realised a tool needed to exist, and we needed to go and build one.



Jason Knight 10:33

Right, so we've got you kind of back into the world of product management, or at least creating products after you managed to escape and are into the ivory tower of venture capital, which I'm sure we'll talk about venture capital again in a minute. But it's just interesting to see you getting called back in like in The Godfather, but the company has been going now for nearly three years, you've got some big customers like T Mobile and Dell, you've got something like 10,000 users, I think, looking at the website, and you've got a freemium model to boot. So kind of going down that product, lead growth, sales motion, but I guess you'd probably sell to big companies, if you could, but a lot of that then starts me thinking about product market fit. And the idea that of course, the goal of any new startup founder is to try and get that and get to that scalable business model and get that growth and maybe even hyper growth over time. So do you feel then with those 10,000 users? I mean, I don't know how many of those are free users versus paid users. But do you feel like you've got really strong solid product market fit at the moment? Or is that something that you're still iterating towards and trying to get some of those people to start to give you some money.



Dani Grant 11:35

we're actually nearing 15,000. It's like any day now, we're super excited.



Jason Knight 11:40

Let's update the website.



Dani Grant 11:41

Right? You know, the best advice I got about how to determine whether you have product market fit is from Mike Adams, the CEO of Grain, which is a call recording tool I love. And he told me that there was a time at Grain where all they could think about was about product market fit. And then there was a time at Grain where they sort of don't no longer thought about

product market fit. And so somewhere in the middle of probably they achieved product market fit. Yep. And I found it was the exact same at gym. So 18 months ago, all my co founder and I talked about all our team talked about was product market fit my phone, even autocorrected OMG to PMF. Like that's how much I was talking about. And at this point, we no longer think about it, we've just sort of moved on to other things. So somewhere in that, we hit it. But you know, you really do need something to measure and something my co founder artefact did so well, and I'm so grateful for is really tracking retention. Retention was the only metric we tracked to know if we were reaching product market fit, he maintained a manual spreadsheet of the users in our alpha. And each week, he would update how many gems they had created that week, and would colour it in if it was more than one, and would not colour it. And if it was zero. And we were just looking for streaks. And I remember the latest iteration of jam the current flavour. The one the one that works. We were we were 19 weeks into this alpha. And there were two teams with 19 week, totally green streaks, and everyone else was pretty green too. But those two teams with 19 week streaks were like, I think this works. And and that that was the beginning of everything.



Jason Knight 13:20

Oh, there you go. So did that. I mean, obviously that took some time to get to that point. But did you find at any point when it wasn't going green that that then helped you to make some decisions about what to prioritise? Or was it just a case of experimenting and just trying different things until the numbers and the lights and the colours all went the right way?



Dani Grant 13:38

We experimented a tonne. When we started the company, the first thing we did before doing anything else was we did 45 user interviews, just to validate that this was not a pm at Cloudflare problem. But this was something industry wide. And those calls. What we heard was so emotional, that and one early interview we did he even tried to pay us and schedule an onboarding for his whole team. And we're like, there's nothing yet. And so we knew we knew this was something to solve. And every single time we iterated and released a new flavour of gym, people would sign up in 1000s. Like the problem resonated, it was just about could we fit the solution to what the market needs. And that sounds easy. And it's hard to do in practice. Like for example, there was one point where we hit a solution where people exactly wanted it. But they didn't have the social capital on their teams to get it adopted. And so that solution doesn't have product market fit. But it is it looks like it could and so we tried so many things. And one of the things that I wish we had done right from the beginning is set that expectation with the team, like when we were early on and doing our first fundraiser. One of one of the VCs that we were talking to told me it's going to take you 18 months from here to get to product market fit and I was like I don't think so. And he was right. It took us 18 months and how does he know that will be As every SaaS company, not every but most SaaS companies that are building a new solution to an existing problem with no product solution, they take that long to iterate on it. So Figma... Figma took a bit longer, but it's also more complicated product Airtable.. Notion took two years. Notion has actually this really dramatic story where two years in, they still didn't have product market fit, they were running out of capital, the founders had to let the whole team go, they moved to a cheaper city, and they rebuilt Notion from scratch. Superhuman took two years, there's an amazing blog post from the founder of being 18 months and still not having product market fit, and how he devised a strategy for how to get there. And

so just the products in our category take that long. And so the advice I wish I had sort of internalised upfront is if you know it's going to take that long, how are you going to approach it? And I think there are two things to do. The first is to communicate that clearly to your team so that when the experiments don't work, it's not demoralising, it's energising. It's like an escape room. It's like, well, what are we going to try next? Versus like, oh, I guess it didn't work. And the second is just throw away code constantly, like you're going to delete everything. And so being more comfortable with that, and just moving fast duct tape.



Jason Knight 16:12

Given that you thought it would take less time than it did, were there any points during that journey that you started to get a little bit demoralised yourself, or were you always so confident in the vision and the problem and the way that that problem resonated that you, you could kind of get through that and just keep on track and not get too downhearted when things took longer than you expected.



Dani Grant 16:32

There are definitely difficult moments where when you start a company, starting a company is a job that grows you up, you know, really fast. Because what you do is you take on a tonne of responsibility both in millions of dollars of other people's money, that they're investing in you versus someone else who could have had that opportunity and the time and effort of your teammates. And they deserve for you to treat that responsibility. With as much earnest and hard work as possible. Yeah. And so you really want to succeed, and you want to do well by them. Because all of these people have trusted you. And so that can be stressful. But with our problem set, we kept hearing over and over and over again, the problems of how we kept hearing such emotional things from people who are trying to use our products, they really wanted us to solve it, they would record these large screen recordings of like ideas for us. And so we knew that there was something there, we just had to figure out what it was.



Jason Knight 17:35

Absolutely. Well, again, it's just important to make sure that if you're confident enough in division that you just stay the course and just keep going right, which is good. It seems that that's really paid off for you. But you talked about fundraising just now. And obviously you used to work a venture capital firm as well. So I guess the first question is, I mean, obviously all companies that get to a certain point, almost all they end up trying to go and get some money either from VCs or from PE firms or anything like that. But you worked for a VC firm. And you I believe that one of your investors is also the same firm as well. So does that mean that getting funding for this was really kind of a gimme? Or did you actually have to put quite a lot of work in to get that funding across the line.



Dani Grant 18:19

We were so lucky with the excitement for solving this problem. And for the belief in us to do so. I have some tips for your listeners who are going out to raise funding. I've seen 100 Plus early stage founders pitch VCs, so I can share like on the VC end, things that stand out

stage founders pitch VCs, so I can share, like on the VC email... things that stand out.



Jason Knight 18:39

Let's do it.



Dani Grant 18:39

So just to level set. I've only... at USV I saw early stage pitches. So this is seed... Series A. And the other thing to level set is... gosh, the VC market right now is tough. And so if you remember there was that like ship that was stuck in the canal, and then that little digger came to rescue it. These tips are like the digger polish. And it still takes a tonne of luck in this really intense market. But the first thing I'll say is one framework to have in mind when you're going out to pitch a VC is you as the founder who is pitching, you are in the business of delivering a great meeting. And once you internalise that helps you make a tonne of the little decisions about how the meeting might go. Like for example, if you're considering flying out and doing the meetings in person or just doing them on Zoom, if you think which one is a better and more memorable meeting, am I very charismatic and I am great in person or am I like do I just have the best mic setup and I'm like, you can sort of internalise that. And you can also think about how do you want to kick off the meeting? Like what's a better meeting? Is it... Hey, where are you based? What's the weather? Or is it telling them about a demo you just saw of this new feature that you're so excited about? Or is it jumping them into a product demo themselves? Like hey, let's just jump in the product right away. And so once you figure out okay, your job is To deliver a meeting that they want to have, again, figuring out what to do during that is much easier.



Jason Knight 20:06

Excellent. So I guess that kind of covers some of the specifics of like how to kind of get off on the right foot, but you still got to actually persuade them to, to give you the money. Now, obviously, in your case, you've got a passion for the problem, you seem to have a lot of validation that the problem was something that people wanted solving. And I guess you then built your whole deck around that. But like, were there any specific approaches that having been a VC or working for a VC yourself that you that you knew what to kind of get you ahead of the game? Like any tips for like, what actually to call out specifically?



Dani Grant 20:42

One tip that Albert at USV, my former boss gave me when we were starting to pitch was investors at this early stage, know that the product will change? They're not evaluating the product, they're evaluating the problem space, and they're evaluating you. And so they're looking to hear how you think about things and whether you've thought about the right things. So often, they'll ask questions, and they aren't listening so much. Like they don't need all the details of the answer, they just want to know that you've thought about it, and you're thinking about it the right way. And so to do that, you can keep your answers short. So often, when we are talking to VCs, we're so excited to share, like, all the details of how we think this could

happen. But a better approach to keep the meeting flowing. And also to show confidence is to say, here's how we think about it at a high level, give like the two sentences, and then say, I'm happy to go into more detail if you'd like and invite them to ask for it.



Jason Knight 21:32

Get off on the right foot and don't over explain, then you'll be ahead of the crowd.



Dani Grant 21:36

There's also one more aspect here about timing. So another helpful framework for when you're raising VC, is to think about when you're interviewing a job candidate. So in this framework, you are the job candidate being interviewed for the job, and the job is getting money from a VC firm. And so when you interview a job candidate, if they you know, months later are still searching for a job, you're thinking, okay, so there, there's no rush, or like, I don't know, it doesn't seem like lots of companies want to hire this person or something. But if you're talking to someone, and they're like, Yeah, I'm doing a bunch of interviews, I'm in some final stage interviews, I'm really thinking, I'm going to make a decision this week, you're like, Okay, this is a great candidate, we want to be the first. And so when you're setting up your fundraising, if you think that way, then what you can do is you can, instead of just starting to do meetings as they occur, you can schedule your fundraiser for a few weeks from now and bundle all your meetings into one or two weeks. And that way, it's like, okay, it's just happening right now. And now is the time versus it feeling sprawled out. The other thing is like, when you're hiring a candidate, if the candidate is not also interviewing you, and is not also thinking about, like, how do I find the best place for me to work, then you're thinking like, oh, they don't have a lot of options, but, but actually, like, the VC firm you choose to work with is sort of like a hire that you make forever. And so you do want to make sure that it's a great fit that you enjoy working with them. And that, that you work together well. And so you can also you should and it shows confidence, ask questions to them about like, how do you work with founders? What are sort of the best case setups? What do you do when things are going poorly and let them also pitch themselves to?



Jason Knight 23:06

So it sounds like your time at a VC firm really prepared you for this, then? And that's then something obviously that really stood you in good stead? So do you feel that there were any good resources that people that maybe didn't have that experience that maybe need to dig into it more like something that you'd recommend that they go and look at or, you know, be it a book, or you know, some TED talk or something that can kind of fill in the gaps for people that maybe like, because PMs obviously these days, as it's become a much wider job, but it's not like they're not all necessarily going to be found as in waiting, yet. There's gonna be people that maybe didn't pick up some of those skills, yet. They're not all out in the valley, surrounded by this stuff, like, Are there any places that people should go to try and find out a little bit more about how to solve that problem?



Dani Grant 23:47

There's a fantastic talk by Rebecca Kate and one of the partners at USP about how to pitch VCs that I will link you to so you can play in the notes. I watched it before we went out.



Jason Knight 23:58

Oh, there you go. But let's talk back a little bit about some of the stuff that you're solving your day job. So bugs are not really just bugs, but the very philosophy behind bugs and this concept, which maybe is falling a little bit out of fashion these days of moving fast and breaking things, you know, the old cliched thing it was it. Mark Zuckerberg said it originally. Now you called moving fast and breaking things. 10 year old advice. That doesn't work in 2023. So do you think it ever really worked or just no one knew any better at the time, and it wasn't enough competition to make it matter.



Dani Grant 24:31

10 years ago, it was great advice. And the standard advice for startups was move fast and break things. And if you ship and you're not embarrassed, you've shipped too late. Yep, this worked at a time where there were not that many software solutions to things. So there were not that many alternatives. And also when people's day to day lives did not live on software. And it was critically important that it worked perfectly every time for them to be able to trust it. And so if you're building especially a productivity tool, it's if it doesn't work, it makes the views are less productive and betrays the promise. So it's, it's sort of just got a work out of the gate. And that was one of those things that we didn't realise early on. But when we realised that it really helped us, one of the things that we did differently about the latest iteration of jam, the one that has product market fit is working is growing is the thing people use when they use jam is, we kept the scope so small, that we could actually ship a bug free product fast, in order to be able to test it, when you're searching for product market fit, what you're actually searching for is clarity. It's sort of not linear, it's not like you're gonna keep adding features until you hit it. It's like you're throwing things out and trying to ingest a firehose of input and data and trying to create clarity out of a mess of data points. And so if users are leaving your product, because it simply does not work, that just looks like you don't have retention, which just looks like you don't have product market fit, and you have no way of knowing are they leaving, because it's not working? Are you leaving, because it's not the right solution for them. And so the biggest service you can do for yourself is, is ignore that 10 year old advice, and actually try to ship something that was and works.



Jason Knight 26:07

So I'm assuming that you don't think that it's ever okay just to crank something out and fix it later. Like, there's no circumstances that you would recommend doing that.



Dani Grant 26:15

So actually, shipping fast is great, because having something public that people are trying to use is the best motivation, you can possibly have to go and fix all your bugs very quickly. So all for shipping fast. But if you really want clarity about whether your your product is working, you

should keep the scope small and the quality high. So you can really test and public what's going on?



Jason Knight 26:35

And what are some of the worst examples, you've seen where that's gone wrong? I mean, obviously, you've worked with a bunch of companies as a VC. I'm sure you're speaking to a lot of companies out there at the moment. They're working on stuff and maybe getting a bit of insight into what they're working on and what they're shipping, or maybe even how many bugs they're reporting. But I just wondered if there was like an example. Or maybe even just a public example, aside from say, seven hours of someone who shipped fast boat things, and it had a very bad effect on their business?



Dani Grant 27:05

An example that comes to mind. I don't know if they moved fast. But do you remember the first time you used Apple Maps?



Jason Knight 27:12

I don't think I haven't used Apple Maps.



Dani Grant 27:14

If you used Apple Maps when it launched, I don't know how long ago, you probably haven't used it again. And that's because first impressions stick. And so it tells you the importance of making sure you deliver a great first impression. But one of the more archaic parts of how product and engineering teams work together that prevents them from shipping great things faster, is that bug reporting process. Most things about how we deliver software today has changed a hundredfold since the advent of the internet, like it's totally unrecognisable, we're so much more in the future. But how we handle and report and communicate about bugs actually hasn't changed since the 90s? It's this archaic, old manual process that slows most teams down.



Jason Knight 27:59

Oh, yes, well, luckily, I've heard that there's someone who's tried to solve that problem. So hopefully, we can go and use that person solution after this call. But you said before this call is every pm and founders job is to support their teams moving fast and shipping awesome stuff. And we just kind of touched on that as well. But before we talk about those PMS and founders in the kind of non developers, what do you think that developers themselves can do to help support the idea that they can move fast, but ship great stuff that works and don't have any bugs in in the first place?

D

Dani Grant 28:29

At our company, we've spent so much time as a team figuring out what works well for us as a remote startup to move really fast. And the reason for that is because as a startup speed is our only competitive advantage against the Giants, so we must figure it out. So we figured out a couple things that work really, really well for us that I think every product and engineering team can implement from the engineering side. The first is building a culture of code comments. So the reason why having great code comments is helpful is when an engineer enters a new part of the codebase. It both helps them ramp up to it faster, but also prevent them from creating new issues. Because they didn't realise the ramifications of what parts of the codebase this code touches. So culture of code comments, every engineer should enforce it. When they're reviewing prs. If they don't see sufficient comments, and they think an engineer who's going to come into this file is not going to have enough context. They should, you know, let the author of the PR Now, the second thing that's helped us a bunch is having an on call rotational programme for engineers. And the way we do on call rotation is different than other companies. So at other companies, typically the on call engineer is doing their normal day job. And in addition, they are on call but at gym, we call thing champion, the champion is only on call. They're not doing anything else. And the reason why this helps us move fast is because that one person is fully focused on fielding all the day to day things, whether it's, oh, there's a bug that we need to fix this urgent or, Oh, this user needs something custom built for them or their server. least that needs to go out. And by that one person being fully focused 100% of their time on those day to day things that come up. All the other engineers can stay totally focused without context switching on whatever it is they're shipping. So that person's one week of being champion helps everyone else ship whatever they're working on faster.



Jason Knight 30:18

Do you think that only works? If they can have a cool name like a champion though, and that they need to optimise the company name for that otherwise, it all falls apart?

D

Dani Grant 30:25

Absolutely.



Jason Knight 30:26

Absolutely.



Jason Knight 30:28

Since we're talking about devs, if I can have 10 seconds of your time, if you're a non technical product manager, or startup founders struggling to communicate with devs, I just like to remind you about the skip level programme, where you can learn to do just that in just five weeks, or without learning how to code. You can check out the show notes for more details. Right... back to the interview.



Jason Knight 30:46

Back to the non developers and the quote unquote business people. What are some of the ways that they can help to contribute to a move fast and build great things environment if we assume that the developers are doing a good job, and it's just down to what the non developers can help to do with that?



Dani Grant 31:02

One of the best things that someone working with developers can do to help developers move faster, is do all of the pre thinking upfront for the developer. So as PMs, we know that when we write PRDs, we're always thinking about what are the edge cases? What are the things that are complicated? What does the developer need to think about, so they just have to write this once, instead of going back to keep fixing slow iteration cycles later. This also applies to bug reports. The reason why fixing bugs is slow and easy bug takes a whole afternoon set of five minutes is because there wasn't enough information upfront. And so focusing on how to supply all that information upfront, either with a tool like jam or by taking screenshots manually of the console logs, the network requests, and writing down all the specs of your browser and device really helps engineers move fast.



Jason Knight 31:49

Now what you've just described, there starts to sound like maybe if I could use a fairly unpopular phrase, these days, there's a little bit of waterfall, this kind of idea or big design up front, you know, you design everything upfront, try and think every single possible thing that could happen and then sort of hand over to the developers and they can get it done. Do you feel that there's any case for for example, coming up with like a collaborative design spec up front? And kind of working it out as you go? Or do you think that there are a lot more advantages from kind of tightening all up before you even get to that development stage.



Dani Grant 32:20

Our engineering team lead, Tony says, weeks of engineering can save hours of planning. I have seen time and time again, the more you can do the planning upfront the the better the project is going to go. Because at the end, you'll just be able to ship and it was to plan, and it's out to users. And then you can iterate later and make it better based on user feedback. But speed is everything. One thing that we've started doing at Jim, which has helped us a tonne is It sounds counterintuitive, but actually spending anywhere from half a day to a full day at the bookends of any project helps prevent weeks of technical debt clean up later. And it sounds counterintuitive, because it's like, shouldn't you be moving as fast as possible. But when engineering and product teams are talking about like moving slowly and slow downs, they're just like the horror stories that they're preventing. And describing is like, the project is taking weeks longer than expected, the deadline is long gone, we keep setting deadlines we can't make this is totally different than we're going to spend an afternoon planning an afternoon of cleanup before and after the project.



Jason Knight 33:25

So what's a quick turnaround for you then if you're spending some of that time building stuff? I mean, do you for example, keep the scope fairly limited so that you can kind of bang stuff out quickly? Or is there like a sort of a cadence that you would push releases out on?



Dani Grant 33:40

We keep the scope small. So small scope, high quality is the mantra at gym. And the other thing we do is we ship everything behind a feature flag so it can go out, we can give it to one user to test tell them it's in beta. And that way, we're starting to get feedback as early as possible or even be able to test as early as possible.



Jason Knight 33:56

Oh, there you go. And where can people find you after this, if they want to find out more about jam or get some advice on starting up their own startup getting some funding or maybe even see if they can get you to go and shoot them in a photo shoot?



Dani Grant 34:08

Well, if you want to try Jam, it's <https://jam.dev>. We want everyone's feedback because we want to bring bug reporting from this 90s process to the modern day. So let me know what you think. I'm dani@jam.dev.



Jason Knight 34:18

Excellent. And hopefully you'll get a few people coming over. Well, I made sure to link that well into the show notes alongside that talk that you mentioned earlier. Hopefully we get a few people thinking about starting anything up or getting some of their own funding. Obviously, appreciate you spending some time talking to us about how not to break things and how to ship stuff better. Hopefully we'll stay in touch but yeah, that's for now. Thanks for taking the time.



Dani Grant 34:40

Thanks so much, Jason.



Jason Knight 34:43

As always, thanks for listening. I hope you found the episode inspiring and insightful. If you did again, I can only encourage you to hop over to <https://www.oneknightinproduct.com>, check out some of my other fantastic guests sign up to the mailing list, subscribe on your favourite

podcast app and make sure you share your friends so you can make sure you never miss another episode again I'll be back soon with another inspiring guest but as for now thanks and good night.