

# One Knight in Product - E171 - Ed Biden

📅 Thu, May 11, 2023 1:37PM ⌚ 37:16

## SUMMARY KEYWORDS

people, fast, teams, product managers, product, badger, hustle, bit, company, release, business, execute, great, set, good, started, learn, guess, games, learning

## SPEAKERS

Ed Biden, Jason Knight

---



Jason Knight 00:00

We've all heard the phrase scream if you want to go faster but what if you want to go faster without screaming? Tonight, we'll talk about how. Speaking of going faster if you want to accelerate your product management career, why not check out My Mentor Path, a new free mentoring platform that enables you to get the support you need to further your career. You can head over to my <https://www.mymentorpath.com>, sign up as a mentor, or mentee or both. Mentoring can be an incredibly high leverage activity for both the mentors and the mentees. So why not give it a try? Check out the show notes for more details. All right, so moving faster, everyone says they want to do it. But there's always something slowing us down like an anchor around our feet. But how can we take off the shackles and move forward with confidence? If you want to find out all about that, stick with us on One Knight in Product.



Jason Knight 01:01

So my guest tonight is Ed Biden. Ed's a longtime product manager, company founder and fractional CPO, who says he's passionate about political reform and once sailed across the Atlantic before Greta Thunberg made it cool. No one tell Donald Trump, now. Ed's a former Christmas tree delivery man who's now helping product managers around the world wrap tinsel and sparkly lights around their roadmaps and getting them set up for the festivities of product management with his new startup Hustle Badger that says that product teams need to get used to executing fast. I don't know what you mean Ed... surely one release a year is just fine, right? Anyway, how you doing tonight, Ed?



Ed Biden 01:32

Really well. Thanks, Jason. How are you? Good. Good.



Jason Knight 01:35

It's good to have you here. All right. So, as ever, first things first, you are the founder, Hustle, Badger. And the word hustle was already making me think of late night coding sessions, five AMI, showers and threads and threads of motivational advice. So try and take me down a peg or two, what does Hustle Badger do?



Ed Biden 01:52

So we help product managers achieve their full potential. We provide templates, case studies and how to guides that basically, we've got a playbook for every situation you're likely to find yourself in.



Jason Knight 02:04

Now, every situation is a bit of a bold claim, because I'm certainly aware, and I'm sure you are from some of the work that you've done some of the situations that product managers can find themselves in, can you really stand behind the phrase, any situation?



Ed Biden 02:16

Well, that's aspirational, I should say, we're working towards it. We're working towards it. We producing a playbook every week. And all of those tables go into our wiki. And so very quickly, we're kind of filling out all the main use cases you've got and things that you would want help for or think, how does someone else do this, then we've got something for you to get started with.



Jason Knight 02:38

But it is fair to say that there are quite a lot of resources and educational programmes and MBAs and mini MBAs and certificates and lovely courses and everything all out there all that good stuff where people can try and do some of that stuff already kind of better themselves and become the best PMS that they can be. So what's the difference between hustle, badger and all of that? And I guess by extension, why is hustle badges approach better?



Ed Biden 03:01

Yeah. So when I looked at the education space, then what I could see was, there was some fantastic training at the top end of the market, and you've got people like product school and reforge. And so I like doing like really fantastic cohort based courses. But they're expensive, you're talking like upwards, upwards, a few 1000 pounds per person to go and do that. And then at the other end of the spectrum, you've got a lot of great free materials, podcasts, articles, you know, there's, there's, that's like, great learning to be had out there for free. But there's very little in the middle, right. And I wanted to provide the kind of consistency and quality that you get with a premium course. But did that add a price point that people could afford to pay for themselves and didn't have to rely on their companies buying for them. And I also have a strong belief that your courses were great because they're intensive, that the best

learning happens on the job, and it's integrated with what you're doing now. So I wanted to create these formats where you could basically kind of like take a template or take a how to guide and you could apply it to what you were doing right now, because that's when learning really sticks.



Jason Knight 04:13

Yeah, I agree with some of that. And I think it kind of chimes with some of the stuff that I've tried to do in previous jobs as well. But I guess one of the questions that naturally springs from that is if we're saying that there's a bunch of expensive courses out there, that I guess you're somewhat competing with, and trying to provide as much value as that how can you do that? Because either these people are charging way too much money, and you know, all sitting there in their yachts, or it just costs a lot of money to do this stuff. And you need a certain amount of money to get the content or get the guides or whatever up to scratch. So how are you able to outmanoeuvre them aside from being new and start up? They How are you able to outmanoeuvre them and still have the quality that you'd need to have to get actually good results out of it?



Ed Biden 04:56

Yeah, that's a fair challenge. So a lot of those premium courses So you're looking at what you're actually paying for his instructors time, and whether that's face to face or whether that's like on a call. And it's synchronous and cohort based, you've still got some really experienced people who are able to give you very tailored feedback. And that is tremendously valuable. But it's also very expensive because those people you need to pay for on an ongoing basis. And what we're doing to bring the cost down is basically tender knowledge for those kinds of people. But document it record it down once. So, yes, you're not going to get a tailored response to your particular question. But most people, like they would take an 80% cheaper products to get, you know, 80% of the value.



Jason Knight 05:46

You're not gonna get ChatGPT to start reading people's answers out there. Maybe. I'm not, though. But that's an interesting point around the production of the content, because it could be that it's just you and your team sitting there kind of slaving away over the typewriter, coming up with as much content as possible and trying to bring your own spin on things and using your own experiences, try and inspire people educate people. But it sounds like what you're doing is maybe a bit of that, but also working with other people getting their content in maybe layering the metadata and the tech on top to to help to surface that to people. Is that a fair shell? Or are you doing more the former than the latter?



Ed Biden 06:24

Yeah, I think yeah, there's, there's layers to this to tell us about your as a as a plan, right. So to where we started is with me writing a lot of the content and putting it online. And it's fairly static, right? Admittedly, it's kind of like it's templates, how to guides that you download, and

you can you go with, and we've started there, because honestly, I was producing a lot of this stuff for my teams, when I've been in CPO roles of VP product roles. And it's been a tremendous help to those teams. And I started thinking, Okay, well, maybe there's a broader market for this stuff, right? So that's not the be all and end all of alpha badger. Yeah, there's really exciting stuff that is going on with AI. So you know, it'd be great if you could have a coach that you could ask questions for, and it would retrieve the right templates, the right kind of sound bites and bits of advice that you needed that moment, even from a kind of a pool of kind of pre recorded or pre documented advice. And I think that's one direction that the latest advances in AI is going to take us is like, who owns those really high quality datasets or banks of advice? And then what are the ways that you can access them? Because it's, it's going to be completely different, right? That's certainly something that I'm really focused on now is finding other really talented operators and capturing their insights and making them available for other people. And doing that in a structured way that makes those those lessons that they've learned kind of like really easy to apply.



Jason Knight 08:03

You're the Blinkist of Product Management thought leaders then or something along those lines. But would you consider hustle badger to be a replacement for some of these types of learning that you're trying to displace? Or do you feel it's more like an addition or augmentation to some of those other ways that you could learn and something that you can kind of layer on top to help you be the best PM you can be?



Ed Biden 08:24

Yeah, another another great question. I think it's, I'm not sure I have a one answer for this or I might, it depends. It depends. It depends. Of course, it depends. Like, I think, yeah, the way I've always thought about hassle budget is if you had to spend money on one thing, you'd probably spend it on us, because I think we're going to be the highest return on investment, you're going to get your time and your money. And then if you do have more time and more money, go and do it premium costs, because like I say there's there's some great people out there doing doing really great work. And if you really want to dive deep into a subject, then you're going to get something out of a cohort based course or even an in person course that just can't be replicated in an asynchronous manner yet.



Jason Knight 09:10

Yet being the operative word, but we'll see I'm looking forward to the roadmap so we can see what stuff's coming up. But you worked at FutureLearn in one of your roles before this as their CTO, or I think VP of product and then CPO. And for my international listeners, that's a British online education platform. I guess you could say it's analogous to something like Coursera or edX. Maybe you're gonna correct me if I've misrepresented FutureLearn, but something along those lines. So you've obviously got form in education, you've worked in that space, and you've presumably learned a thing or two in your time there. But what was it that made you want to start your own thing in that space, rather than maybe go work for someone else's doing it themselves? Aside from the approach, I guess, but like, it's still a bit of a commitment to set something up yourself, right?



Ed Biden 09:57

Yeah, I think yeah, I've always I've been obsessed with education since I was I was really lucky with my education, I got a fantastic education. And I went to Oxford. My first job. My first job was at McKinsey. And I felt that I learned more in three months in McKinsey than I didn't three years at Oxford. And that was a bit of a, you know, I was like, why is this? But yeah, it kind of got me thinking as like, how can it be that one of the top universities in the world like doesn't really teach you what it's like to be in the workforce? Now? Maybe I chose the wrong degree, or maybe I didn't apply myself sufficiently like, I'm not gonna say it's all Oxford's fault, right? Like, there's there's a degree of like self ownership that, but I think since that point, it's got me thinking about is, you know, has education kept up with the way that the workplace has changed? And so I've always been thinking about how we can we can solve that better. But yeah, you mentioned Future Learn, I think, yeah, great, great description there. That was my first really in depth time working in the ed tech sector. And I loved it, I've kind of like felt like that was my opportunity to go and sort of ask all these questions that I've been asking myself for years, but actually, maybe get some of the answers and dive a bit deeper and see it in a professional context. And a lot of what I found out there and so that hypothesis, I was testing I've, I've obviously carried forward into in Hustle Badger.



Jason Knight 11:29

Yeah. And I think it's really interesting. And it's something that I've chatted to other guests about in the past people working in the tech space to the idea that education is, in some ways ripe for disruption, right? Because there are so many for want of a better word like legacy attitudes, or just old ways of doing things, and especially, maybe more in the actual public education space, where you're actually looking at, you know, like government sponsored education and stuff where they're very averse to change. Whereas I guess with you, you're in a situation where you can kind of go out there and you've almost got a blank canvas, to some degree. I mean, there's obviously certain norms and ways that people learn, but you still got a way to actually go out there and try something different. And yeah, maybe hit the spot or maybe iterate towards that. Do you feel it's kind of nice to be in less of a structured, but of existing thing, so you can kind of just swing for the fences?



Ed Biden 12:18

Yeah, I mean, there's, there's definitely something very exciting about starting something from scratch. And for me, that's always been an attraction. As a third business I've I've started from scratch. Neither of the previous two have been particularly successful, I should say. But,



Jason Knight 12:35

You know, third time's a charm.



Ed Biden 12:37

There you go, you know, the speed with which you can change things and speed with which you learn. And seeing some things have started to make an impression on the world, which would not have existed if you were not there, for me is just an incomparable feeling. So yeah, I really enjoy it. It's not it's not all good. There's some stresses as well. But I think that's definitely the benefit I keep coming back to.



Jason Knight 13:00

No, absolutely. I think it's always important to sort of see the impact you're having on people as well, right? Like, it's really nice to see the results of trying to make people's lives better in some little way. So yeah, absolutely agree with that. But do you feel that your experience at FutureLearn and the experience that you had there, building those types of education products taught you either things to specifically try to do or maybe even things to avoid in the current approach that you're now championing, like any kind of takeaways that almost informed positively or negatively what you're doing now?



Ed Biden 13:31

Yeah, I think yeah, when we were at FutureLearn, there were some some core hypotheses that we were testing, things like subscription, personalised learning, which for you were experimenting with and testing. And they weren't quite right for for FutureLearn at that time, but I could see the value of them. And I could see how they'd be really powerful in a different context of a different business. So some of those things have been like quite central to how I thought about what Hustle Badger could be going forwards.



Jason Knight 14:02

There you go, you can start to do all the things you wish you could do at a time. But how did you learn how to get good at product management yourself? Like, obviously, you've been in the game for a while now. So you've learned a lot on the job, I'm sure there's a certain amount of school of hard knocks, but there wasn't a hustle badger before you started it. And you started it after you started being a PM. So were there any kind of key ways that you or key resources that you used in the past or approaches that you took to your own self education that either did or didn't inform what you're doing now, but in any case, kind of ways that you would recommend to kind of up your own game?



Ed Biden 14:38

Yeah, I mean, the first pm role that I had was a games company in Paris that Dylan Sang who is CP... or CTO at Bumble. She was working there. I vaguely knew her and she said, Hey, can we work for this company? It'll be awesome. And I sort of jumped at that chance both to work in the games industry working Paris, but also to work with him who already done a number of years of PME. At that point, I learned from her and that really got me started. And from there, I then went to work for Wooga in Berlin, another games company, but they were quite established, by the time that I joined them, there were probably 200 people. And that was the time that Facebook games mobile games were just kind of going crazy, he was using his

heyday. And so product management was being practised by those companies quite a high level. And at the same time, it was one that an ex McKinsey consultant like myself could sort of slot into fairly easily. Because a lot of those games were simulation games, they had very complex economies. And I was used to modelling those kinds of modelling businesses because that's what you do as a business analyst as a strategy consultant. So that was kind of my bridge and my angle in and it got me my toehold. And then it was just like working with other people and developing some more of the kind of the softer side, the design side that I'd neglected for a few years.



Jason Knight 16:11

So very much sounds like a school of hard knocks, or maybe not always hard, but kind of a very much on the job, sort of see and observe and copy and become so which, frankly, is the kind of way that I like to approach it as well. So I'm all for that. Speaking of becoming, if you'll give me a few seconds just to reach out out to my mentor path, a new free mentoring platform where you can sign up to be a mentor, mentee, or both. Try to help you and your peers, be the best selves and become the best product managers they can be. You can check the show notes for more details. All right back to the interview. One of the things that you learned what you said that you learned in your career, and we chatted about before this, and something you're particularly passionate about, is the idea that product managers and product teams need to learn how to execute fast. Now, we can speak about what fast means in a minute. But however fast fast is, why do you think it's so important to execute fast?



Ed Biden 17:11

I think it's really important to execute fast, because all things being equal, executing fast means delivering more impact, you get more shots on goal. It's faster learning, and it's faster, the cost correct. And I think sometimes product managers get a bit hung up that Oh, well. You know, if we're if we're shipping just really fast, we're not really thinking about what we're doing. And I think that's kind of the wrong way to think about it. Because the fastest you're shipping, the more that you're learning, the faster you're going to understand what you should be doing. And there's always an element of uncertainty to what you're doing. So it's better to embrace that uncertainty, get something out in the world, see how people react to it. And then, you know, adapt, and course correct from that point.



Jason Knight 17:59

So, so far, so lean startup in a way, like, that's not a completely controversial opinion, you know, test and learn and validate your learning and all of that good stuff. But I've certainly worked in some places in the past where maybe there was almost maybe an appetite for things to slow down even from the customer side. Like they didn't want things too fast. Like we were working with banks, or we were working with big companies, big enterprises that they kind of wanted stability, or like working with APIs, and people don't want the APIs changed all the time. Do you think it's possible to be too fast?



Ed Biden 18:30

Yeah, I mean, I think I think you're right, like there are definitely customer groups where they're not going to appreciate it if the API structure is changing every week for sure. Like one of their their core needs as a user, right is going to be stability. So then you're going to have to think about, well, what are the other user needs that you're trying to serve, while still respecting that needed, they have continuity and stability and their business security and so on.



Jason Knight 18:56

So yeah, I think I agree with that. And this idea that it's about being fast, or as fast as is appropriate for the circumstances that you're in. And this is something where maybe we start to clash a little bit with the Agile fundamentalists out there that are sitting there saying that you need to do your 1000 releases a day and all of that stuff, like you're just like Amazon, and like, Yeah, that's cool. And all but again, you've got to kind of adapt to your own circumstances. But in your opinion, then, Minister find fast. What does fast actually mean to you? And I guess by extension, what's the fastest you've ever been? Like... what's the fastest you've ever managed to execute on a sustained basis?



Ed Biden 19:29

Great question. I think I think it's really difficult to know what fast means. And that's going to mean different things for prepends different businesses, right? Are What is your tech stack look like? Or what is the maturity of the product because a literal day one, it's the first day you've sat down at your computer startup is going to build things at a different speed to someone working in an incumbent business that's been there for 20 years in a regulated industry. For me, you know, I've had the, I'd say like the privilege to work at Rocket Internet at its heyday. And Rocket Internet users back in 2016 2017, had a standard goal of launching a new business in 100 days. And this is before you've got things like Shopify and platforms like this to kind of like set up new businesses. And I set up a business in 28 days from meeting the engineers who were going to work on it, today being alive for customers and taking bookings. So that felt fast. And yeah, we were talking about how customers were going to get the emails confirming the order they'd made. An engineer told me it was gonna take three weeks to build the email system. I was like, that's not gonna work. We're gonna launch next week.



Jason Knight 20:47

Typical product overreach. There you go. .



Ed Biden 20:49

Yeah. So you know that that I mean, it was great for understanding if you have to work to very, very short timelines. And if you really want to move quickly, what kinds of trade offs and what kinds of hacks and workarounds Can you can you make to go at that speed? Because there's, there's always trade offs, right. And I never look at it as like, this the is just as fast as we can go. And oh, if we tried a bit harder, we could go faster. That's never the case. You're always trading something. It's just a question of what you're willing to trade, or whether it's worth it.





Jason Knight 21:18

Yeah, reminds me of the episode of The Simpsons where Homer was sent, or he moved away to some managed community to manage some super villains nuclear power plant. And you're well done if you've seen the episode, but he kind of walks in and he's told to motivate the team. And he's like, What are you working? And they're like, yes, yes, sir. Mr. Simpson? Yes, we are. And then he's like, could you work a little bit harder? Oh, sure thing, and they start typing faster. That doesn't normally work. And there's that whole mythical man month thing going on as well, you know, you can't just make people work faster, you've got to try and be more efficient, or try and handle scope and all those things. But some people might think, hang on a minute, whenever I've been told to execute fast in the past, it's always been from the founders or the sales team, saying, We've got to release stuff, we just got to get stuff out, we got to be seen to be being innovative. And they start to pull the feature factory alarm bell, because they start to think they're just being forced to just push stuff through a breakneck speed. And I know you touched on it a little bit about some of the things that you might want to think about. But how can we ensure put maybe guardrails around the fact that whilst we do want to deliver fast that we're not delivering any old nonsense, or just basically pushing stuff out for the sake of it?



Ed Biden 22:30

But firstly, I mean, that's the sort of like why the founders ask those questions, because founders and CEOs always want to go faster, right. And I agree, I've seen that so many times, where the people at the top right, yeah, but we should just be going a bit quick ish and weird, it just feels a bit slow. And usually what the reacting that way, too, is they're not getting enough transparency from the product teams about the complexity of the challenges that they are facing, and the trade offs of the product teams are making. Because maybe the CEO is super happy for things not to be 100%, scalable, or 100%, you know, secure for a limited period, whilst we test something, what they really just want to see is how it's performing the market. But that's a decision that you can kind of expose to the CEO and letting them make, you don't need to make it yourself. And when you start doing that, you start getting a lot more buy in from CEOs and CEOs tend to slow down a bit, because actually, they realise that you are making some quite sensible trade offs about risk. And maybe they don't want to be GDPR in compliance. Future or whatever it is, right. So I think that's kind of the core of it is like, can you be really transparent about what is driving the timelines, and give options that allow people to change those timelines for different things that you can trade off, or whether that's completeness of the feature sets, or stability, security, like things that you might consider as an engineering team to be kind of absolutes. But actually, a CEO who's managing like, lots of different risks at a macro level might be willing to kind of flex to some degree.



Jason Knight 24:11

Yeah, I think one of the key phrases that you caught out there was this kind of for the time being, like, you know, we'll make it not scalable for now, or we'll make it not quite perfect for now. And, you know, that's something which sometimes rings alarm bells for people, especially when they've been in situations before where they never got to go back and fit and finish the job, right. They, they kind of just had to move on to the next thing and the next thing and what

happens or what can happen is that you end up in two years time with a complete hodgepodge of stuff because you never went back to maybe mitigate some of those risks. So I guess there is a danger for that. But I guess that's the job of the product team to keep on top of it.



Ed Biden 24:51

But yeah, 100% I think that's so important. Like if you're asking teams really to move quickly, then you can go faster sometimes by incurring a little bit of technical debt. And you have to give them the time to catch that up otherwise, that is going to accumulate over time and get you to a position where you're slowed down whether you like it or not.



Jason Knight 25:10

So what are some of the other barriers that you've seen that prevent teams, I mean, that's not just pick on the product managers, but like the entire product development teams, the trio's or just organisations in general, but some of the barriers that maybe stop people from moving as fast as they could.



Ed Biden 25:27

So I think there's two things in particular that I see occurring a lot, and one is a lack of focus. So it's very common for product teams to be dealing with not only their own roadmaps, but incoming bug requests, incoming requests from other teams. And that can really reduce the amount of time that they've got to make progress on their strategic priorities. So if you don't have that kind of real laser focus on getting one thing done, then it's very easy for you to be distracted, find your focus to fuse and make very little progress against anything. And the second thing I'd say is that it's not enough to know the direction you want to go in, you've actually got to understand how you're going to get there. So you have to have some fundamental insights onto how are you going to solve the problems that users have that allow you to get to the business outcomes that you want? So you need to know, what problems do our users have? And what are the sorts of things that we can do to solve those problems. Again, otherwise, you're just sort of shooting in the dark, and your chances of making any meaningful progress towards the business objectives that you've set yourself is negligible.



Jason Knight 26:41

Never happens. But sometimes the problems that can cause this, to be a problem can be a little bit outside the PMs, or traditionally, the PMs remit, and almost permeate the entire culture of the company like the engineering team, they just want to work in a certain way. They don't believe in certain types of building software, or they've had bad experiences in the past and never again, well, there's an aversion to showing MVPs to customers, because our customers are too high value or everything has to be a big bang release, because we need to kind of line it up with marketing. And like, you start to lose some of that ability to do some of that fast iteration and getting stuff out and testing and learning. And that's, you know, that does happen. But I think it's fair to say that not all PMs are directly responsible for all of the things that they'd have to do to fix some of those cultural problems within a company. Maybe they

just have to work via influence, to nudge people in the right direction, with data, evidence, principles, whatever it is that they can use. But when it comes to nudging people in the right direction, and trying to make some of the changes, maybe that are outside of the product team that are needed to execute more quickly. What are some of the levers that PMS can pull, or some of the approaches that they can use to try and sort of unblock some of those things?



Ed Biden 27:55

I think what's really important is to understand your stakeholders really well, firstly, so you've got to get beyond understanding the features that they want, and really dig into why do they want those features. And once you start understanding why they want those features, you start understanding their worldview, then you're going to be able to have a much better, more collaborative strategy that involves both products and non product initiatives. And those can then work in concert. I think a second thing that makes a big difference is really understanding who has decision rights and who's accountable for different things. So as a product manager, you should usually assume that you are the person that is accountable to things going on, because that is your role to make things happen. And you should attack ambiguity. If people don't seem to be responding to you acting as the accountable person, then just directly ask them, hey, is this something that you think you should be taking care of? Again, once you once you do that, then it's much easier to go fast. I mean, the ambiguity of ownership is such a killer to velocity, because everyone spends so much time dancing around trying to figure out who should be doing what, who's got the final call, does this need to be consensus based, and actually, just to have one person who says, Hey, I've heard all the different points of view, and we're going to go for this option. It just cuts through so much, so much stuff. And the final thing I'd say is, make sure you've got regular check ins, because I see a lot of teams start to drift when they stop having regular progress check ins about where they've got to, and where of the other teams that they rely on where they got to. And when you have those those progress meetings, they don't need to be long. They don't need to be exciting, but just having them and having them publicly means that you're talking about what is important on a regular basis. And that that keeps important things front of mind for people. And that means that they spend time working on what's important, I think choice can be such a powerful tool as well. Yeah, this is not a problem for product managers alone, right? This is a, this is a team sport. This is something where a product managers really need to coordinate a team response, the interspeech their team needs, they need to speak to their designers, or really have a very open conversation about how can we go faster. And it's really what retros are for, in my opinion, there's, there's a time and there's a place to celebrating people. And that's, that's important. But more than anything, your retro should be about asking the team. Honestly, how can we do more in the next cycle? With the same number of people with the same objectives? How do we do more? How do we go faster? What what is stopping us now? And recognising those obstacles so that you can tackle them?



Jason Knight 30:42

No, absolutely. I think that idea of almost relentless improvement. I mean, on the one hand, I absolutely agree. On the other hand, I guess it could also feel a little demoralising, I guess we do have to have some level of that kind of cheering about what we just shipped as well, right. Because otherwise, people start to feel again, that they're just in this kind of constant march into the future and never getting stopped to take a breath. So trying to get that balance, right, I

guess, is really important. But on the understanding that you might not want to give away too many juicy details or trade secrets, are there any examples from your career where you've maybe executed really fast or as fast as you could made a big splash and that you almost put it down to the speed that you executed as well as what you executed?



Ed Biden 31:26

Yeah, I'm really proud of what we achieved at future than when I arrived there. And I think within two weeks of me joining, then, the company as a whole kind of decided that it wanted to launch a new business model, and a new learning model combined. And previously, when we done that sort of project, it had taken about 18 months, and understanding the work that was involved and looking at the scale of the teams. You know, and and obviously in conversation with the other, the other kind of technical leader, an engineer and design leader as well, we set ourselves a goal of delivering it by Christmas, which was four and a half months later. So we set ourselves this incredibly ambitious goal of moving four times faster than we historically had. And then we put in place the systems and some of these tactics that I've been talking about. They're very clear lines of accountability, regular check ins, radical focus on on that particular goal of getting that live. And what was really nice was that when we came to towards Christmas, everyone was in you know, about to head off for the holidays, we went and asked the teams and we have five teams working on concert on this, whether we should wait until January, the first which was our agreed launch date to release or whether we should do that before the holidays before everyone went away. And the teams voted on that. We gave him the say on that. And they voted to launch before Christmas. So we not only went really fast, but we also ended up releasing ahead of schedule, and in a very low stress way that the teams controlled, despite the fact that they hadn't been the one setting the original deadline. And then outcome wise, it was important to go that quickly because we essentially reached a point where we were becoming convinced that subscription was the way forward and to test that we needed to get a subscription product to market. And I think within just a few months, we had 25% of our revenue coming through subscription channels. And that that that was like a huge proof point of the hypothesis that we'd had.



Jason Knight 33:34

Oh, there you go. Well, that was kind of put no deploy Fridays into context where it's almost like no deploy Christmas is the ultimate no deploy Friday. Right, then what's your position on no deploy Fridays? I mean, it's been in the socials a little bit recently, the kind of the pros and cons of being able to just release code whenever you want. You shouldn't be afraid of deploying anything on a Friday, it should be just the same as any other day and and other people like, well, you know, you could wait a couple of days or get it out on Monday so that everyone's in the office, like where do you stand on that?



Ed Biden 34:00

I mean, I think I stand on that in the same way I acted a feature they speak as a team's she would have they think that what are the risks that you're you're trying to avoid by not releasing on Fridays, because there's no one it's interest for things to break over the weekend. You don't

want that as a leader. You don't want that as a team member, right? So if the teams are happy to release on Fridays, and I trust them to release, and if they don't feel comfortable with that, then as a leader, I think I could be happy that when it's a Monday, it's not a big deal.



Jason Knight 34:30

So actionable advice time we've got a pm listening to this maybe stuck in a slow moving organisation, maybe failing or not getting anything done. They'll be stuck in analysis, paralysis or stuck behind whatever it is that slowing them down like a organisational lobby in front of them going up a hill on a motorway or highway from my international listeners. And they want to hit the accelerator or the gas pedal, and they want to start going faster and we've talked about some of the big picture things you could do but like what's one concrete little step that they could do not to because it's Friday on Monday to start going up through the gears.



Ed Biden 35:04

So I'd say for one really tactical piece of advice would be to start tracking how your team is spending their time, and what percentage of time you actually spending on strategic planned work versus on everything else. Everything else, like what are you spending on ad hoc requests from other teams, bugs that come up, updating the tech stack, you know, you're updating the modules, you're running, all that other stuff, and see how much time you're really giving yourself to act on the things that you think matter. Because for a lot of teams, you'll probably find that less than 50%. And then obviously, if your study less than half your time on the stuff that you think is the most important, then you're probably not going as fast as you want to be.



Jason Knight 35:50

Oh, there you go. Excellent advice. And I'm sure there's gonna be people listening to this connect, go faster stripes on and the aerodynamic helmets and all of the good stuff that I need to do to make sure they're streamlined as much as possible. But where can people find you after this? If they want to chat about hustle, badger or find out generally speaking how to go faster, or maybe even see if they can get the inside track on a good deal on a Christmas Street?



Ed Biden 36:13

Yeah, well, Hustle Badger is the obvious place to find me <https://www.hustlebadger.com> or LinkedIn is fairly easy to find me. Ever since there was a president with my name, then.... Yeah, I tend to turn up in search results.



Jason Knight 36:25

That's SEO for you, I'll make sure to link that all into the show notes anyway, and hopefully, you'll get a few people speeding in your direction. Well, that's been a great chat. So obviously, really appreciate you spending some of your time to help us understand how we might always

bid faster, and maybe even teach yourself a few things along the way. Obviously, you and I'll stay in touch but for now, thanks for taking the time.



Ed Biden 36:44

Thanks very much. Good to come along.



Jason Knight 36:49

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