

One Knight in Product - E140 - Allen Holub

Sun, 9/4 7:14PM 51:54

SUMMARY KEYWORDS

people, agile, organisation, scrum, product managers, backlog, developers, product, problems, jira, scrum guide, job, work, talking, twitter, blockers, company, point, coaching, agility

SPEAKERS

Jason Knight



Jason Knight 00:00

Hello, and welcome to the show. And in the words of Steve Ballmer. Tonight, we're talking about developers, developers, developers. You know, those people actually make the products and seem to spend half their time on Twitter shouting your product managers, but can't we will be friends. Speaking of which, if you're an old friend or new, and you haven't done it yet, why not pop over to one night in product.com? Take a look around, sign up to the mailing list or subscribe on your favourite podcast app to make sure you never miss another episode again. So yeah, product managers and developers, we all work on teams and organisations but how can we work together more effectively? And come to think of it? How can we make the organisations better while we're at it? If you want to find out if shouting at people on Twitter that all this stuff actually works? Keep listening to one that's important.



Jason Knight 00:56

So my guest tonight is Alan Holub. HoloLens is an internationally recognised Software Architect, author, Coach and Trainer who wants to help you build software better and build better software, islands, all about the agility and development practices, and none of these tasks and frameworks that might hold you back. Now I have to say I thought I was into the agile mindset, but Alan makes me feel like I'm in a bubble going over a waterfall. Alan's also active on Twitter with a punchy, opinionated style that has led me to put my gumshield in boxing gloves on as a precaution tonight. Hi, Alan, how are you tonight?



01:25

Thanks. Hopefully, there won't be any severe injuries.



Jason Knight 01:31

I was gonna say there's only two rules, no biting and no punching below the waist. So if we can

keep to those, I think we'll be fine. So before we get into some of the meaty issues that we're gonna be discussing tonight, I want to find out a little bit about you what makes you tick, what you're up to outside of Twitter, where a lot of people know you from right now your bio says, I help you build software better and build better software. So what types of people are you helping? And what are you specifically helping them with?



01:56

Uh, you know, it's companies all over the map, all having to do with Agile stuff, I use the word agile, with some trepidation nowadays is



Jason Knight 02:05

big IO little a



02:07

little late. I'm trying to focus on agility and all my marketing stuff, because it kind of gets the idea across, but it's not the dreaded a word. You know, and instead of agile, Andy hunt said Agile has come to mean do half of Scrum badly and use JIRA? And I think he's right about that. So the way that I think about it is that I help people solve problems using Lean and Agile thinking, which is a different thing than, you know, agile with the quotes and little trademark after it.



Jason Knight 02:38

But you could try them out your approach as well, if you want. But I guess coaching is an interesting one, because you are a coach, or at least partly a coach.



02:45

Well, I guess I do, we can discuss that what that means what that word means. I'm certainly a consultant, I can help people, I give them advice, I point out problems, I help them solve those problems. Coaching, Alyssa Atkins is that I think none of us is service with her book on coaching, because everybody read it. And it's all very, it's so touchy feely in places, it's driven the coaching community, even more in that direction. Right. And it's the I've just seen a lot of people that call themselves coaches that what they're talking about as life coaching. They're like coaching, certification boards, and they have one of the certifications, you know, and they say, it's my job to help people live their full potential, and I'm going no, not as an Agile coach, that's not your job, it's your job to help them be more effective in producing software. And, you know, and that involves a certain amount of kind of coaching technique in the sense that you can't just order people around, right? If you're going to be effective about this, you've got to sort of teach them the pluses and the minuses and give people a palette of things that they can choose from. And it's it's not my decision when somebody needs to do it's your decision. Yes.

So I see it as my job is providing you with the information that you need to make a good decision. Whether you want to call that coaching or not. I don't know. A lot of people use the word but I don't know what it means.



Jason Knight 04:06

Just to confirm there are no certifications with your name on that you're selling to anyone at this moment.



04:11

There are no



Jason Knight 04:15

Well, you know, never say never vote, you know, if you ever need to get a new guitar in the background or something, there's always that,



04:20

you know, I did a lot of teaching for the UC Berkeley Extension or University of California extension, and they gave out certificates, but they were certificates of attendance. You went to all the classes and I'm happy with that, right?



Jason Knight 04:34

Because we're gonna measure something you measure that you write certificates



04:36

as to mastery I have a hard time with. I can't vouch for somebody being a master if I don't work for them, or work with them at least.



Jason Knight 04:44

But how much of that work then as you kind of just touched on where you're going in there giving advice given your experience, giving them some pointers in the pros and cons, all the stuff that obviously you do in your day job you talked about on Twitter, but how much of it is like you going in there? Doing all of that doing some session was whiteboards, whatever, some group sessions throwing the Batman smoke grade on the floor and flying off and letting them just deal with it? Like, do you have any kind of ongoing relationships with these people? Or do you kind of just why?



05:10

Yeah, I prefer to have an ongoing relationship, it depends on the company, right? The jobs that I've liked the most, I've had ongoing relationships have lasted at least a year or so, with the company. Sometimes they just want me to come in and teach a two day class and get out is that I'm not cheap, right? None of the really good consultants are cheap. And though they probably wouldn't have any problems paying my rate to Deloitte, they have hard problems paying my rate to me. So I don't know what that's about. But on the other hand, the amount of money I can save a company is huge for you, right? So it's everybody who's competent. And



Jason Knight 05:45

I'm trying to think I'm competent. And I'm just judging by what I read, I'm gonna put that on my gravestone.



05:53

But yes, and the you know, the, the amount of money saved is always much higher than the fee no matter how high the fee is, if the purpose is a good person, right? They're doing the job 100%. But I think the biggest problem is that of the things I do, right, some of it is just kind of nuts and bolts. How do you do stuff? Classes? How do you put together a user story? In fact, I'm doing a public class on user stories in a month or so? And how to come up with an agile friendly architecture? And what are the mechanics of event storming? What are the mechanics of test driven development? That kind of stuff? Right, just



Jason Knight 06:27

got to be TDD, right? It has to be TDD. I mean, that's your thing, right?



06:31

Well, TDD in my, I extrapolate TDD a little bit further than other people do. Right? So this TDD is Kent came up with it started out with about with this small thing for developing the code that's in front of you. Yep. Right. And then Dan North came along and made it a bigger thing. Right, and sort of brought in all of his acceptance criteria and business criteria and that kind of stuff. And I'm, I'm taking it a step further, all the way out to something that's even more abstract and more general. And then, but that doesn't mean that I give up on the stuff in between, right? I do all three. Yeah, so I'm, I'm an outside in kind of person. And I like to start at the, at the domain, and work my way into the stories using TDD, inspect and adapt and do a little work and run some tests, rather than that kind of thinking that theory, but it encompasses pretty much everything. But the point is, is that if I'm really going to be successful at making the company, more agile, to use the dreaded a word, the Agile with a lowercase a, who people have to be talking to at the sea levels, right? Are they the people that are really in charge of the corporate culture? And 99% of the time that people who find me and like me and want to hire me? Are they the engineers, not the sea levels? Yeah. And that disconnect is hard, right?

Because they're in a position where they don't have to sell my services for me to a sea level, who's dubious about whether he or she needs them. And it makes for an interesting problem, right? Because often, often, I'll be brought in to do one thing and end up doing something else, just because people understand that it's useful. You know, and I don't know any way around that particular problem. I wish I did from you. I wish I did. But I don't I don't know what. Well, I



Jason Knight 08:09

was gonna say, though, because you touched on it, just there. I mean, it's long been a suspicion of mine, that the vast majority of executives, in any company, from the most agile to the least agile, probably don't really care too much about how the sausage is made, as long as the sausage is made, right? Like, that's a disappointment for people like you. And to some people like me, but



08:30

the thing is, is that what this, I can certainly help people with making sausage. That's not going to get any agility into the organisation. Right? Yeah. So that's the problem, though, is that they'll bring in somebody like me, who, with the intent of teaching people sausage making, when really what I'm saying is, do you really want to be in the sausage business at all? is right and the, the, that disconnect is sometimes frustrating is the sometimes it's fine. Sometimes I end up sort of moving up through the organisation and making an impact. And I like that. Sometimes the person on top will be some kind of tyrant who? Well, one of the most unpleasant jobs I've ever had was working for a company where the people I was working with were great. They were spectacular. And the people who were running things were some of the worst people running things I've ever seen in my life. They were tyrants. They held the people who were doing the work in contempt, they would sneer and roll their eyes and talk about learned helplessness and all of this stupidity, right. This wasn't learned helplessness, this was actual helplessness is that the people who were treating the people who were doing the work with such contempt, were causing huge numbers of problems. But they made it very clear that if I brought those issues up, the work was done. I was out of there. And they in fact, did fire me once. Then reluctantly brought me back and then fired me again. Well, because I wouldn't shut up right there. That's no no you Alan. We're not hiring you to do lighting people. We're not hiring you to get people to think we want you to teach them test driven development and if you escaped from that silo of teaching test driven development, you we've had it with you go away. And you know, and they were a sort of a fake scrum shop getting more and more fake by the minute and the time that I was there, they had migrated away from agile in several critical ways. And they just didn't want help. So you know, you are,



Jason Knight 10:20

there you go. So what kind of percentage of firms do you reckon actually do kind of engage with you at that top level and actually enable you to make some proper organisational change rather than just teaching developers how to make sausages more efficiently.



10:31

among small companies it's very high is the most of the places where I do the most good are the startups in the sort of up to medium sized organisation, which is, you know, maybe 1000 people. Once you get above that the resistance to being agile gets stronger and stronger and stronger to the point where it's just, it's not worth it to me to beat my head against a brick wall, I'm just not interested. So if it was a really big company, I got called up by the CEO, and said, Can you help us and we had a talk and they were committed, I'd be all in favour of doing that. I would love to have another job like that. I'd have them occasionally and rarely, and I really enjoyed them. But if I'm like the enemy, I'm not interested. And if what they want is for somebody to come in with an army of trainers and force everybody to march in lockstep to a canned process. I don't do that. But that's often what they want. So there's kind of the tension there.



Jason Knight 11:27

Well, speaking of tension, as mentioned earlier, he also very visible on Twitter. Some might say that you're, in fact, some people have said that you're somewhat outspoken. I know some people have said that they don't particularly jive with your style. Some people even said that they've stopped even following you at all, because they just don't really want to read that stuff anymore. So you've obviously got a reputation. I guess I have to ask, is this all part of a facade that you're putting on to some extent to just drum up business? Like, are you naturally punchy? Or are you a bit of a teddy bear? Really?



12:00

I'm an introvert. But people don't know what the word introvert means. You know, if I, if I want to relax, I don't go to a loud party. But I must say, almost all programmers are introverts, introverts, we're introverts and really means is that you're able to focus. And if we couldn't do that we couldn't be developers, right is that we're all introverts.



Jason Knight 12:20

Just for the record, before you start to cosy up to be too closely, I must say, I've not been a developer for a few years now. So you, you can have at me if you want



12:28

still. It's the mindset, right and 100%. But the way that I'm looking at it is that I just kind of got tired of walking on eggshells all the time. And a lot of the community coaching community, that's all that they ever recommend. Oh, my God, you can't say that. You can't just say stuff here, because people will then push back as hard as you're pushing forward. Right? And I'm crying. I'm not pushing, I'm just speaking my notion of what the truth is, in a direct way. Yep. And if you can't handle directness, that's fine. But I don't see that I should need to preface everything I should say with? Well, in my opinion, it seems to me that maybe in some situations that might help a little bit, if you had thought about doing it like this, right. And I just don't see any point in that is that the the if people want help, I'll help. And if they don't want to help them, I'm not going to help them. And I don't see why I should pander to the people who don't

want help. I pander isn't the right word, really. But I don't see why I should. I don't need to spend a lot of energy, trying to convince people of stuff. Diana Larson said it pretty well. She said, I'm not in the convincing business. And I kind of agree with that. I'm not in the convincing business. So if you've already been convinced, and you want help, that I'm happy, but I'm not going to convince you to be agile, or convince you that agility is a good thing. Or convince you that waterfall command control stuff is not a good thing, because I'm just not interested in doing that. It's too painful. Yeah,



Jason Knight 13:56

I get what you mean, I think one of the questions that arises from that is whether you feel that that kind of limits the impact that you could make either via your consulting directly or via the visibility that you get from Twitter, because of course, you could go out there, be a little bit more meet people in the middle and maybe helped to change a few more minds by making them feel that this isn't quite such a big change. Do you? Do you feel that you're risking just sort of preaching to the choir if you're being so kind of



14:23

No, because I can focus? I can help them with details that they don't understand is the there are plenty of people that are doing what you're saying. Yeah. And I and there are very few people that are just coming in and being straightforward and honest with people and I would rather be that person. And the, you know, I'm gonna do what makes me happy. I'm getting old enough at this point that I'm not willing to spend any brain cells make me happy anymore. And having to having to walk on eggshells in order to not get fired is not my idea of being happy. So the I've just want to do it and the the thing is, is that if somebody He knows that they need to change in some way. And they just want help with details. You don't really want all that dancing around the edges, stuff, you want straightforward advice, and somewhat execute on that. And there are enough people out there in that state that I think I can help them. You know, and the people that follow me on Twitter, mostly, not all but mostly agree with at least the basic premises that I approach things with. And, you know, you would say there are people that don't follow me anymore, because I don't know why. Right, most of those people fall into the Craig Larman. Camp, right and lemons laws camp. The basic idea of LeMans laws that he plays is that he's saying that the point of most organisations is to maintain the status quo, right, and they'll do anything that they can to maintain the status quo and lahrman standard law in particular is the one that I think is important, because as soon as you start talking about stuff, they'll start saying, well, in the real world, we can't do that. And that's a fantasy. And that's theoretical, and that's ivory tower, and that they would know, that's never gonna work. And really, these are all just just smokescreens meant to hide the fact that you think that what you're doing is just peachy, and you're not about to change. And if that's the situation, fine, I have no problems with that. If what you're doing is working really well for you, you should absolutely keep doing it. Right? I haven't I have no problems with that at all. But a lot of the people who fall into that category, what they're doing isn't working particularly well for them. They just don't want to admit it for one reason or another, or they can't, right. So some of it is just training is the you look at all the MBA programmes and stuff, and people are trained that a business must do X and income the actual people and they said no, oh, no, who you really thought that through. Maybe we don't want to be doing X, maybe we're gonna be doing something else right estimation, is the one that you know, the big one that comes to mind immediately. And, you

know, the, if you get somebody who has the idea of if we did what you suggested, we wouldn't even be a business because businesses don't do that. Right, and that I don't want to make fun of those people's they've been brainwashed, admittedly, at least it seems that way to me. But it's what they believe, to be dumb for me to say you don't believe what you actually believe. They believe that and that's fine. But I can't make much of an impact in that kind of situation.



Jason Knight 17:22

So we need some kind of intermediary type sort of someone in between you and then that can kind of transform them enough that they are prepared to take some of your ideas on board is that what we're saying we need some kind of Demi alum, I don't think that'll work



17:35

either is that you've got to get people to understand that they need to make some changes if they want to improve. And some people don't, and I don't I don't see that there's any strategy or something that you will use to trick people into suddenly being changed or not being changed versus that. Either are they aren't. And if there's a way to get people to change that, which is kind of a personality trait. I don't know how to do it. I'm not a trained psychologist, I you know, I have no idea how to do this kind of stuff. So I, I don't concern myself with them. Because I can't I can't help them.



Jason Knight 18:07

That's fair enough. But I guess then that's talking a lot about some of the organisational muscles, the instincts, the biases and the opinions that some of the people within the organisation have up to and including the management in the executive suite. But you've obviously also got developers in these companies. And some of these developers, I've seen replying to some of your tweets, maybe pushing back on, not necessarily the idea of being agile with a small ad or the idea of doing some of the stuff but maybe pushing back for example on well, we don't want to do tests or test slows down or what do you mean, you shouldn't use databases in a microservices or all of the other things that kind of come up from time to time? Because, you know, we all know that developers can be opinionated. Right? But you know,



18:46

again, people have to make their own mistakes. And you know, somebody saying, Well, I think we should have a big, centralised database with microservices, they'll learn pretty quickly why that's a bad idea. Right? And what I can do is I can try and head off some of that pain and say, it's probably not gonna work, here's why. And here's something that will work better. And but I'm not going to I'm not going to tell him what to do is that that's, that's how agile would that be? There you go. You



Jason Knight 19:17



Jason Knight 20:17

can't micromanage a man and you



19:19

can micromanage people is that you what you can do is provide information and let people make their own decisions. And that's fine, right. And a lot of times, though, what I get in the way of pushback is not we don't think this is a good idea, but rather they won't let us do it. Yeah, whoever they is. And will they them is sometimes the team itself, right? But yeah, but often it's not right often the team really is good and they really know what they're doing. The company I was talking about a few minutes ago right with our tyrannical leaders, the the people in the teams knew what to do. But they were in an organisation that was ordering them around and punishing them deeply if they didn't follow orders. And they were afraid there was a lot of fear in that organisation. And, you know, I remember one situation where me and a colleague, without being scripted, sat down and did an hour long presentation on what user stories are and how to actually deal with them. And they flew into a panic. They were saying, Oh, my God, what you're telling us to do is not what the head honcho chief agile person told us what to do. So it's wrong, everything's wrong, right. And they were like, running around like chickens with their heads cut off in a panic for two days. And not only that, they were also in a panic, because this the scheduled half hour meeting went half an hour over. And and, you know, it was discouraging. It was discouraging. But nonetheless, even those people understood the difference between things that work really well and things that they were permitted to do. You know, and ultimately, in that particular organisation that being permitted overruled the doing what I think best. And that's unfortunate, but there it is.



Jason Knight 21:06

Yeah, well, it does happen. But you've recently said, The best way to undergo an Agile transformation, going to the small aim is not to get coaches and frameworks in the light, but just remove the blockers to agility, and then the rest will just kind of take care of itself. Myself, that's what I was gonna say, You better give yourself and then I'm gonna go off the rails. But does that actually really happen? Because I can understand where you're going with that one. And obviously, if people want to do it, and they're being blocked from doing it, like you just say, then obviously, you take the chakras off, and they can take off. But I also have this image in my head. And this is at least partly based on past experience of my own is where sometimes you feel like maybe you're working both with developers, and to some extent with product managers that have maybe never worked for a particularly well functioning company. And none of them have got any instincts, or their muscles or anything in that area. And if you just left them to it, and took away the blockers that kind of just sort of sit there and probably do it roughly the same way, because they've never seen it done better. Like do you think that it's optimistic to think that it's gonna happen like that



22:12

one of the primary blockers is not being a learning organisation? Yeah, one of the primary blockers are things that are in the way of people learning. And those can be anything from bureaucratic blockers, like, you've got to get permission from your boss who has to get permission from their boss who has to get permission from their boss, and every penny has to

be justified and Yetta, Yetta Yetta, right. Oh, this, this, these insane processes that cost more than the classes that they are not letting you take because you're too expensive. Right, so there's that problem. Yep. But it could also just be a culture, you know, the other blockers to learning is not having time to learn. Right? So removing the blockers doesn't mean that you leave people adrift. It means that you remove, you have to remove the blockers carefully in a way that make sure that people can learn and become more agile and lowercase sense. So starting with the two things that I usually start with are trying to get some scope for learning, setting up a learning environment. Last time I was brought in, as a interim consulting architect, for a company, first thing I did was I set up a quote university, so that we could all have at least a few hours a



Jason Knight 23:31

week. Sounds like I sent diplomas, awesome certificates, and so on, well, but



23:34

it was just here's how, you know, I did a bunch of lectures. And then I kind of ran out of stuff to talk about, and I started getting everybody else involved in the lectures. And after a while, it was just going by itself, right? So the people in the organisation, were teaching each other how to do things, which is how it should work. Yeah. Right. So that's eliminating a blocker, though, right? Because they're the blocker there was that people weren't teaching each other. So we're trying to set up things so that people can teach each other and that's, that's a good thing, right. The other thing that is usually a blocker is getting getting the work under control. So it's an important first step. And the blocking thing there is this notion of we have to have a big upfront plan and follow it to the letter. And you see that in backlogs that have 10 years worth of work in them, and which continue to grow, right. And that kind of we have to have a plan, right? So those again, are the sorts of things that are creating friction that you need to work on. So you can say all right, let's make the backlog have a month's work kind of just let's remove some of that friction. Right because that's a blocker to change there is the fact that you feel compelled to do everything in the backlog in the order it's presented. that's preventing you from being agile because you're afraid to add anything because I would just make the backlog even bigger. Right so it's a process you but it's not just just stop everything you know is wrong. Just stop doing all of it. And I'm done now, so I'll leave right and leave all this chaos behind you is I I'd see any value in that particularly.



Jason Knight 25:02

Now, I think that thing about backlogs is really interesting. And I know that you're in favour of very small or no backlogs yourself. But at the same time, I think even when people do have backlogs, there's almost this implicit expectation from maybe the rest of the organisation that maybe have visibility on that, that eventually some of this stuff's gonna get done, when we all know, you know, I know every developer in the world knows that probably 90% of the tickets on that backlog are never gonna never, ever, never gonna get done ever. Like there's 0% chance of ever going to get



25:30

done. And that, which is to say, you're just lying yourself. Yeah, it's just



Jason Knight 25:33

a comfort blanket, right? So you can say that you've got that feedback back, and we'll look at it one day, but that they literally never comes.



25:39

So address that. And there are lots of ways to address it, you can make the backlog smaller. Right? If if you really want to go for shock and awe approach, you can just throw it out. Just today, so just throw the whole thing out.



Jason Knight 25:55

Right? How many times you've done that though, Allen, come on. That sounds like the sort of thing that a few people I've done it twice have the guy. I've done it twice. And



26:01

it's been very successful both times? Well, usually I get too much resistance to get them to throw it out entirely. Yeah, I can imagine. But a lot of companies that I work with don't have backlogs. If you started when I started with the startup, we can set up a process that works off the bat. Yeah, right. The biggest is fun. I'm, as you probably know, from Twitter, not much of a scrum fan. And I don't want to. But the thing is, is that Scrum is not I go back and forth about Scrum is that for a knowledgeable team, I think of Scrum as mostly harmless to steal a phrase from Douglas Adams. Oh, yeah, it doesn't do a lot. It doesn't do any good at all, but it doesn't do much damage. On the other hand, if you're just starting out, and you, you do the scrum thing. And that's the first thing you've ever seen. And that's the way you define agile and it sets up. I don't know how to say it, but it kind of gives you a mindset of this is the way we must work. Yeah, then it is doing active damage that point. And what started me on this is that concept of backlog, that's a scrum thing. That's not an agile thing. notion of that backlog does not appear anywhere in the Agile Manifesto. It was not really part of any of the early Agile processes Scrum. There's a great propaganda machine backing the scrum industrial complex, where they try and present themselves as the first and best and only agile process and that's all complete BS. You know, Scrum, Scrum, the origins of Scrum go back a while but the origins of everything come back a while is that we can we can say the origins of agile really are in the 70s with the Toyota Production System or even further back to that, right and to say Scrum is the oldest process because of that, you know, the new new product development game paper. And you know, that's a great paper and what they're describing has nothing to do with Scrum least not as Scrum, not scrum as present. And you know, to argue Jeff was doing Scrum in 1998. Well, Kent was doing XP and 99. All of us were doing something agile back then, before the word agile existed. The actual history of Scrum is that XP was dominant for years. The problem with XP is that traditional management doesn't much like it doesn't have it doesn't have that wrapper of bureaucracy around it that management likes. So scrum came in,

relatively late. I'm just guessing here, but I would say 2005 plus or minus. And with all of their certification and insanity, but the business people that wouldn't accept something like XP, because they saw it as chaotic, would accept something like Scrum, because there was some structure there. So what they were selling was a structured wrapper around agile. But the more structure you had, the less agility you have. So it kind of got things started in the wrong direction and things like backlogs, this notion. Well, we're agile. So we have a backlog and I'm going to No, that's a scrum thing. That's not an agile thing. We were agile for 10 years before scrum came along with its backlogs. And no you don't need a backlog. You don't need a PIO and you don't need a scrum master and you don't need a sprint and you don't need any of that garbage. Right. And the people get locked into thinking that it's somehow mandatory.



Jason Knight 29:21

I'm sure you're more of a release train type of guy anyway. Right?



29:24

It depends on how you define release trains. And it's certainly not in the safe sense. Well,



Jason Knight 29:28

I'm talking about the safe one. But you know, the notion of a release



29:31

train came out of Spotify, and they do a pretty good job of it. The basic idea of we're gonna release every Friday, regardless. Yeah. And if you don't want people to see what you're working on, put it behind a feature flag. I can get behind that. I



Jason Knight 29:47

see nothing wrong with that. Me too.



29:49

I'd like to release more often than once a week. But the point is, is that your code is always releasable. Yep. But if you don't want somebody to see it, you hide it behind a flag. That's I see nothing wrong with that as well. Have something at all, that's fine with me.



Jason Knight 30:02

No, I love that as of last week. This is also fair to say, though that I mean, it's easy for us to dig into scrum because there are so many poor scrum implementations around the world. I also

think though, the one thing I mean, I agree with you, by the way, I'm not I'm not a massive scrum fan. I, I know that lots of people use it. It's kind of okay, but at the same time, I don't live or die by it. But I also do get quite close. Sometimes when I look at people digging into it and slating it when they then describe something which even the scrum guide didn't say to do, like it's just some random hodgepodge of Scrum and waterfall and micromanagement and other stuff that they just kind of squelch together. And then they sit there say, Well, that didn't work very well. Scrum sucks.



30:44

Well, yeah, there's a lot of that. But an agile tool, not just Scrum, but agile. But the same process. But the thing is, is that I know a few scrum trainer types, mostly with Canons organisation, and I like them as people and I, I respect a lot of them. A lot of they're really good agile people, they really know what they're doing. But they've kind of jumped onto the scrum bandwagon, which sort of, I think makes them less effective to be honest as, as teachers and stuff, but the, the point there is that they're always saying, but that's the bad scrum people. And if you do it, right, you do the good scrum stuff. It's great. And I look around me though, and I'm going so many people are doing it wrong. Maybe the problem is not that there's all these people that are not doing it, right, but rather that the framework itself is flawed, right? There you go. The it's more basic than people will



Jason Knight 31:40

purposefully incomplete. And remember, purposefully incomplete, like that helps,



31:45

right? Or to put it another way, even if you do it perfectly, according to the scrum guide. It's not possible to be to do it, right. You know, when I bring up with a lot of these people, I will bring up the ambiguities in the scrum guide of which there are many. Oh, yes, we will say one thing and appear at the beginning of a paragraph. And we'll say something else at the end of the same paragraph. And the honest people I talked to kind of get these sheepish expressions on their face and go oh, well, yeah. But what it really means is I'm going but if you have to resort to what it really means is, then you failed as a process. Yep. Right, is that people are doing Scrum, thinking that they're doing the right thing. And yeah, they're reading the scrum guide. Yeah. And they're doing what they think it says, And they end up doing this crazy stuff. And maybe the problem is very basic to what Scrum is. And you go even further back, though, right? And I think, well, at the core of agile thinking is this notion of a self organising team. And the notion that you give people the support that they need to get the work done, and then let them do it. And yep, if you're going to be agile, then you've got to be able to have the agility of being able to define your own processes, right? You've got to define processes that work for you. So at the core of Scrum, is a refutation of that. It's like no, you can't define something that works for you. You have to have sprints, and you have to have sprint reviews, and you have to have daily scrums, and you go through the litany. Yeah, so the core idea of a framework. And this is not just scrum as any of them. But the core of it, the idea of a framework flies in the face of agile thinking, right? And the way the way to be agile is to learn and to come to some decisions and make some decisions and to develop a process that works for you.



Jason Knight 33:35

Now, that makes sense. But you mentioned it just now. And you've mentioned it a bit on Twitter seems to be I mean, you've kind of already touched on your distrust of micromanagement. And to some extent management in general, you also seem to have a bit of a dig at product owners and product people in general. And now this, this is going into my world now, Alan, so we have to be very careful if we can argue about something. We can argue about something? Well, I was gonna say like, I mean, obviously, I understand what you mean about wanting to do things in a properly agile way not micromanaging, not being too prescriptive and not being too dogmatic. And I think one of the most interesting things that I see online is a lot of people maybe have dealt with quite poorly performing product managers that are probably working in quite poorly performing organisations. I guess, at the nub of it. Can we be friends, as product people, and developers? Or do we have to always be shouting at each other?



34:28

I don't think we need to be shouting at each other. I don't think there's anything good coming out of that. There a couple of ways to say this. First of all, I have nothing against product management. You saw here first, product management as a skill. And a specialty is essential. And that product managers do things that desperately need to get done, which developers will not do if left to their own devices. Right things like market analysis, right is that somebody's got to do that. Yep. And coming up with a VIABLE experiment. So in order to verify ideas, product managers are that's their job. That's what they do. Yep. So in any conversation about developing the software, having a voice in the conversation, it's bringing those perspectives is, is essential. Goes beyond invaluable. It's essential. So I have no problems with with that, right with the role of Product Management. But where we might? Well, it's not really but but I think the first one we might differ is how to integrate that then to the organisation and into the process. Yep. I don't believe in having a siloed product organisation. And I don't believe even in having a named product or organisation if the way they say that is, I'm one of the product people. And I'm working with the team. Right, what I want to see is teams that have product expertise on them. People the product, people will say, Well, yeah, but we've got to get together as product people and talk about product sometimes, then I'm going Yeah, sure. Absolutely. I did no problems with that. But it's not because you are the product team. And that's your that's the basis from which you work, but rather that we're working on the development teams. Alright. So in other words, when I think about a product team, I don't see any difference between product team and development. And that was the Spotify guild system comes to mind immediately, as we're talking about, we're doing it right, I said that I think that's great. Right? So product, people have to get together and discuss product, things that span the teams occasionally. So there has to be some alignment mechanism inside the organisation that allows that to happen. But I really disagree with the idea of there being a centralised product organisation that's making those kinds of decisions which are then diffused out into the organisation as a whole. And I really disagree with the notion of a product owner who makes all the decisions. That's just a single point of failure. As far as I'm concerned, I don't want to have single points of failure in the system.



Jason Knight 36:55

Well I mean I disagree with product owners in general to be honest as a concept I think that

well, I mean, I disagree with product owners in general, to be honest, as a concept, I think that what's happened and I blame safe for this, to some extent is product owners have been kind of, they're now being hired as product owners, like that's the job title on the job description. And they're being hired very specifically to just jockey the backlog and move stuff around and tell the developers what to develop next. And, like, there's obviously always a bit of that when you're talking about prioritisation, and, you know, explaining what it is that's important and where we should be going. And I think that is a product managers, particular skill set role, or whatever, right contribution to the team. But just having someone whose only job is to sit there moving and backlog around and write tickets, it was like a real where it starts to blend and bleed a little bit for me,



37:40

I think was joopa Hill, who called them ticket monkeys is today. I might be wrong. I apologise that Jeep if he did come up with that, but I love that term. Right. But we don't need that. Yeah, absolutely. And which brings us back to backlogs, right, that whole idea of a backlog and tickets and all of that craziness has nothing to do with agility? I don't think it helps. Right? And if you look at you look at the original idea of user stories, right? The work that the Ron Jeffries in Hendrickson and Kent Beck were doing in the original XP team. Story was a very small thing. Some of you could write on a on an index card run register published to Twitter, a picture of his desk, which has got a bunch of yellow stickies stuck on it, right, those are historic. Those are his user stories, a sentence or two on a sticky, that's enough to remind you that you have to have a detailed conversation later. And those are easy to manage and easy to get rid of and easy to easy to do everything with because there aren't any details there. That has turned into this monstrosity, these Ticket Monster problems, right where people have these story will go out to pages and have detail after detail after detail and you got to estimate it and then re estimate and then re estimate the estimate. And it's just insanity as far as I'm concerned. So the product management getting back to product management. If we're talking about stories in the original Hendrickson, Jeffrey's back kind of sense organised on something like a product map like a Jeff Patton style product map or product magnet story map runner. Yeah, that's great. That's exactly what we should be doing. Right? The whole idea of a user narrative and coming up with stories and see how they fit into the narrative and trying to go you know, product use flow and all that kind of stuff. That's all really valuable stuff. We should all be doing that. Yeah, but beyond that know, people that get jobs as POS you will see them in those job descriptions facility with Jura is always a primary requirement. You know, and no, I'm sorry, is that that's that has nothing to do with being agile that has to do with having an upfront waterfall plan. And you can call it whatever you want. You can call it a backlog, but if it's a point, if it's a waterfall plan, it's a waterfall plan. Calling it a backlog doesn't change



Jason Knight 39:59

but you you explicitly have blamed Java before for driving some of these practices as well. Do you think that's fair?



40:06

Yeah. Because people don't know better is that they, you know, they get a CSP or something. And then they get their first job as a green Po, there's never done any work and they don't

really know how to do product work at all. And they're told well, your job is to manage these JIRA tickets. And then they learn that's the way to do it. And then you do that with a whole organisation. Bring in the army of Scrum trainers and get everybody to do Scrum. And nobody knows any better. So JIRA. It doesn't control but it's it strongly influences the way they work because very opinionated juries, you tend to do the things that make JIRA happy rather than doing the things that make for effective product development. And it doesn't work because it's not well,



Jason Knight 40:46

yeah, I think also, you get the wonderful situation sometimes where there's only certain people that can do certain things in JIRA as well. And everything kind of gets micromanaged at the permission level within JIRA as well. So no, you got you got to speak to Dave over there, too. If you want to change the story points on a ticket or that sort of thing. Yeah, but



41:02

you shouldn't, what do you need? I don't know what you need any of that for? You look at JIRA. And what does it do? First of all, it does surveillance. Yep. And we should not be working in a surveillance culture. Secondly, what it's doing is estimation. But estimation makes no sense at all at the backlog level, because as you just said, You're gonna throw out 90% of it. Fingers crossed. So what possible value is an estimate that is based on something where 90% of it is not going to ever be implemented? Is what what have we achieved there? Right. And it encourages you to collect too many details too soon. Right. So there's just a lot of stuff that it's kind of encouraging you to do that it's not helping. And it would be best not to have it, you know, every the most effective organisations I've ever worked in, including the ones that I've been CTO for. We had sticky notes up on the wall. That was it. That's all we needed. Yeah. And now that everybody's working with notes, we have sticky notes in Murrow, but it's still just sticky notes. It's nothing fancy. And that's all you need. And the argue otherwise, is to argue that you need that the plan is more important than the people. Individuals is what they say, individuals and interactions over processes and tools. I don't really believe quite that anymore. I wouldn't use the word individuals, I think we need to be working in a collaborative unit. But certainly people in their interactions is more important than any kind of process or tool. And what's happening is the tools are dominating. And the process you're we're a scrum shop, and we use JIRA and we do Scrum. And so here we have a process and a tool. And where you're talking about people interacting in that process. Yeah, right. And I don't hear it right is the emphasis is on the process and the tools rather than on the people



Jason Knight 42:48

and specifications, ie



42:51

specifications, and schedules and deadlines, and just all of this craziness, which is the norm under a waterfall phase gated situation, and their core concepts that people hold on to with a death grip and are unwilling to let go. Agile is all about letting go of that death grip, and it's

scary. And it's not it kind of throws people into a world that is unfamiliar. So they don't like it. So they lash out against it. And that's understandable, I do nothing. You know, it's really it's a reasonable reaction, all things considered. But to argue that Agile is just this thin veneer of, you know, Scrum Masters and stuff put on top of what we're doing now that sort of misses the point.



Jason Knight 43:38

So get the Sharpie, put two week lines on the Gantt chart, and you're done. Yeah. But in your view, what is then the best use of a product managers time during the development process. So week by week as the team doing their thing they're building out and executing against the overall vision of the company doing it the way they feel best. And using all of the whatever approach that they want all of the stuff you kind of generally recommend, I guess, but what does the product manager do for you in that,



44:09

let's talk about a few of them. One of them is, in an ideal world, you would be releasing to your actual customers, maybe multiple times a day, and getting feedback from your actual customers and adjusting what you're doing based on that feedback. Most of us, no matter how well intentioned we are, can't do that. We don't have that kind of daily or hourly contact with an actual customer. So we have to have somebody who speaks for the customer. And I think a product manager is a good person to be doing that. So I think one of the most important things that they do during development is answer questions. Yep. Then one particularly bad organisation, I just enforced a rule on them. I said if you have a product, if you have a question for the product manager, they must answer it within two minutes. I don't care what meeting they're in. I don't care where they are, if they're talking to this CEO, I don't care if a question comes up, you've got to answer it in two minutes. And that did a huge amount of good, you know, and we, the rule only had to be in place for a week or so. But once people got the idea, then they couldn't imagine not being that way, because they were, here, you've got a product manager saying, I can't be bothered. And then the team, a team of eight people goes off for six hours, and does the wrong thing, and then has to throw that work away. There's no as possible economic justification for that. So the way that they can do the thing, though, is that they cannot see themselves as an intermediary between the customer and the developers. So the other key thing I think the product people have to be doing is they have to be facilitating direct communication between the developers and the customers, and setting up meetings so that they can talk directly to each other not acting as an intermediary. And then during that those meetings, and during the whole development process, the role of the product manager is basically to constantly be harping on, does this fit into our company strategy? Does this fit with the needs of the market? Does this fit with, you know, all of those kinds of product questions? And you've got to have somebody that's working with the team pulling things into into place, if you will, they're not ordering people around, they're just asking the right questions more often than not. But it seems to me that a dysfunctional product owner is one who orders people around, who says we are going to work on these three stories, this sprint, and if you don't like it, I don't care, right, we don't need that. And I see a lot of that. But I see a lot of a lot of push. I'm a big fan of pull systems. And I see a lot of push systems centering around Scrum and product owners and that kind of thing where the product owner is basically assigning work to the teams. And that's not the way it's supposed to work. Even in the scrum guide. That's not

the way it's supposed to work. But you still nonetheless see it a lot. So the product owner has very little to do with the actual collection and refinement of stories. That's down at the bottom of the list. Because the stories come from the customers, and the people who need the details about implementing them, which is to say the details of the customer's problems are the developers. Right? So what the product manager is doing is getting those two parties together so they can talk to each other. And then providing the input that the team needs that neither neither the developers nor the customers have, right and knowledge of the entire market, for example. And so the input that they're providing is invaluable. It's essential, that strategic input. But as soon as as soon as they become surrogate customers, things start falling apart is that they can't it's not the way it works.



Jason Knight 47:42

Yeah, no, I think I agree with much of that, and probably don't have time to pull it all apart. But I think one little bookend on that, though, is and you'd be horrified to know I'm sure and probably completely unsurprised that in some companies, the product managers themselves don't even get to talk to customers. So that's



47:59

what is that then? Well, well, you know, back back in the dark ages of 30 years ago, there was product management as a profession, right, as business analysis is what it was usually called back then. Yeah, they saw it as their job to design the product basically. And then then they had to convince the team to implement the thing that they invented. So a lot of the job was a persuasion. You're trying to persuade the team to implement this thing that you came up with as a BA. And those days are, fortunately over in the Agile world, the actual Agile world, but they're not over in the world as a whole. Right, is that it's still a lot of, there's a lot of thinking along those lines. It's still going on, and I don't think I don't think it works.



Jason Knight 48:46

Well, no, I definitely agree again, with most of that, and hopefully we can all be better product people as we go. But where can people find you after this? If they want to find out more about? Well, I've no test driven development, or map their path to being truly agile? Or maybe if they just fancy an argument.



49:03

I can give you a whole long list. There's Twitter. Yes. Right. So I'm at Allen Holub, A I, L E, N, H, O L, U B. So that's probably the best way to just sort of get into a quick conversation. I do do public classes is going to my website is holdup.com. In fact, I have a class coming up in about a month. I don't know when this is going to come out. But I have a class coming up on story creation and management that's focused actually on both product owners and developers or product product managers and developers. So I'm going to cover from both sides so that it will be in the end of September. We will be out by then. Yeah, we'll be out by then I do this periodically. I'm writing a book. Oh, wow. The which people have been asking me to write a

book for like, a couple years now. And I haven't been willing to do it. But I'm now writing back kind of inspired me and I'm stealing his ideas that he said, we're going to be agile about this. So what he did is he set up a substack and he said, Okay, I don't want to hear from everybody, because that would just be chaos. So I'm going to put this little minor \$7 A month subscription in the way of everybody being able to chime in. So if you're interested enough that you're willing to bundle up seven bucks a month, you can contribute. And he's doing the book incrementally one chapter at a time getting feedback, adjusting based on the feedback, he gets that kind of stuff. And I'm saying, Well, great. That's a great idea. So I set up a sub stack that's called no book. Is that the title of his hashtag? No. And it's basically about all the things that we've been talking about and things that people do, in my opinion, wrong, and how can we fix them? Right? Obviously, a book that was just about to do, people did things wrong would not be of much value. So it's mostly about fix it. But anyway, I have this I have the notebook thing going and eventually it's going to be on Leanpub. Okay, good. So, you know, all these projects, but there we go. And of course, if you want to hire me, hire me. If you go to holo.com/chat, you can set up a video, call them we can talk about what you need.



Jason Knight 50:56

Oh, there you go. I'll make sure to link that all into the show notes. And hopefully you'll get a smattering of new twitter followers and maybe even some teams can come and hire you and give you loads of shiny coins to go and make them better



51:07

for shiny coin assault and demolish a gross shiny coin.



Jason Knight 51:13

Well, that's been a fantastic chat. So, obviously, really appreciate you spending some of your valuable time telling the rest of us what you think and how it should already work. Hopefully we stay in touch but yeah, as of now, thanks for taking the time.



51:24

You're welcome. My pleasure.



Jason Knight 51:27

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 [white knight in product.com](https://whiteknightinproduct.com) Check out some of my other fantastic guests sign up to debate Mr. Subscribe on your favourite podcast app and make sure you share your friends so if you are making never miss another episode again. I'll be back soon with another inspiring guest but as for now, thanks and good night.

