Jon Kern (00:04):

Even as a developer, working on a feature, you have to understand it ain't real unless it's out there and the customers are using it.

Dan Hardiker (00:11):

quite often, I find organizations spend a lot of money hating the product when really they hate the process. And if you blame the wrong thing, you end up throwing it out and saying, let's replace JIRA with the next new shiny.

Jason Lopez (00:23):

There's a way to collaborate, to make software, which goes a little like this, you and your team and how you work together is more important than what kinds of processes and tools you use. It's better that the software you're writing works rather than just signing off on documentation and throwing it over the wall. You involve your customer for their feedback along the way. They're not an adversary and you're not neurotic about sticking to a plan, but adapt and respond to changes. That's the gist of the agile manifesto. It was written by 17 software engineers in 2001, their purpose was to codify just how teams ought to collaborate, to write software. It's often been said. The reason for outlining an agile process was as a pushback on the methodology known as waterfall in which there's documentation and sign off on each phase of a software project. The proponents of agile felt that software teams would do good to their companies as well as their customers, by being more responsive. And so one feature of agile is delivering value to customers in smaller frequent deliverables.

Jon Kern (01:32):

Do everything you can to reduce the gap in time from doing something to getting feedback.

Jason Lopez (01:40):

John Kern is one of those 17 engineers who met in snowbird, Utah, back in 2001 to set down the agile manifesto today, he's an agile transformation consultant at adaptive list.

Dan Hardiker (01:53):

It's not about replacing waterfall. It's actually about the core principles of agility and bounce guessing change into people's hands quicker.

Jason Lopez (02:02):

Dan Hardiker is the chief technology officer at Weaver, which makes software for architects and the construction industry. For this story. We interviewed John Kern and Dan Hardaker together. Ken Kaplan of the forecast is the interviewer in this Q and A. Both Kern and Hardiker are evangelists for agile methodologies and talk frequently with teams about how to implement it. And they also talk to people already doing agile and how it can benefit a company's whole culture.

Dan Hardiker (02:30):

I'm trying to expand the scope from beyond that team and give them empathy of what they're asking about the business, the business, wanting to know what do I get, when do I guess it comes to going to cost me it doesn't really relate to engineers that have to deliver something every two weeks. And so having that empathy for how you're asking the rest of the business to change, it's not just you that has

to change. It really helps them understand the fear that they put inside the organization and how it's that job as well, to help the organization adapt. Not just that. What I try to

Speaker 2 (03:00):

Encourage teams to do is use a systems theory and be holistic. You're not this little Island of isolation within the business, or even within the application. So it's, it's having a broader perspective. It's again, it's, it's being able to be slightly above what your, where you're actually working so that you can make informed decisions and from a more holistic point of view. And that's why I love what Nutanix is doing. I mean, I'm jealous that I don't have that capability, that Nutanix as, but they've really, they're really pushing agile and lean concepts under the guise of making things, uh, you know, run anywhere, making everything work together, they all together now phrase, right? I mean that, to me, that's, that's an idealistic view of, it's not a bunch of separate little things working now. You're, you're part of a system you're part of a whole.

Speaker 2 (03:52):

And even as a developer working on a feature, you have to understand it ain't real unless it's out there and the customers are using it. How do you know when you're not being agile? Is it a, is it a feeling, is it a reaction? How do you know you're not being had down? You gotta think, Oh, that's a good question. Typically, I'll try to ask 'em teams what it takes for them to deliver. Good friend of Dan's that, uh, I had the pleasure of meeting works at a very large bank and he did a great exercise cause I would always ask, what would it take to deliver the smallest change? And I'm thinking like a typo, you know, it's just something trivial he asked, what would it take to deliver no change, just go through the process. So it kind of starts with having enough humility to question, how are we working? How do we, you know, can we do better? It's a little bit like two second lane. You should be able to come in every day and think of something I can do better and be more lean and be more, um, efficient and effective. That's probably the hardest thing is for people to not fall into the trap of just come in and every day working like an hourly employee doing nine to five, it's very challenging to be agile and that you have to constantly evaluate what you're doing. So it's not easy to relax.

Speaker 3 (05:11):

I'd say that if you're worried about agile, whether you're doing agile properly, or the things that hold you back from doing agile, it's going to be the ceremonies where you do stand ups each day, or you're doing your backlog sprints. And when you get to the retrospective, is this the retrospective that we do every two weeks and what comes out of it? Is it the same stuff that we had last time and the time before are we just covering new ground? How have we changed scrum or the agile processes that we're using from three months ago or six months ago, if we're doing it the same way we were a year ago, that's half the incrementally improving. I mean, we might be improving our software, but we're not improving how we deliver software. And that was the point behind that job was to constantly look at you and how you do things and give people the pragmatism and the empowerment to make change. And if you're not changing your processes, you're only changing a software. You've just changed how you deliver software. Not really fundamentally changed to an agile way of working.

Speaker 2 (06:09):

And sometimes we'll see organizations who will be going through the ceremony. I was just talking to a gentleman at a bar last night and he described their retrospectives and their ceremonies. And I don't want to say they were all wrong, but they were basically all wrong. He was talking about how the

retrospective wasn't about how we work, what we, cause I would say, you know, you could even say something like the pizza was bad last week and anything, but he said it was a ceremony that just the product owner and the scrum master, arguing about the backlog and nothing about the team. So that there's unfortunately a lot of agile in name only or foe agile. And then agile gets a bad name. Um, and adaptive is we're experts in confluence and JIRA and all things that Lassie and, and JIRA often gets a bad name too. It's it's a similar thing. It's not the tool, right? It's not it's the tool might be a start to help you do something, but it shouldn't be the end. Just like scrum might be a start, but it shouldn't be the end.

Speaker 3 (07:12):

There's a saying in scuba diving that tourists come along, especially tourists that have money and they have all the gear, but no idea. And now the dangerous ones, because they're the ones who buy the advanced equipment that can get them into situations beyond their control. And when you're 30, 40 meters down with a tank of knots in your back, you can start swinging down the up. And if you're giving JIRA, then quite often I find organizations spend a lot of money, hating the products when really they hate their process. And if you blame the wrong thing, you end up throwing it out and saying, let's replace JIRA with the next new shiny. But you just take all of your ways of working, which is what people really hate from the tool to tool. And you think that technology will save us when actually it's in our hands.

Speaker 4 (07:55):

It sounds like, uh, with this systems and teamwork's working together, is there a, uh, a handicap from trying to document more things and have more data to help make your decisions more data driven? Is that bogged down agility or does it,

Speaker 2 (08:12):

It's a great question. Even the way you're phrasing, you know, handicap, but if you look at the agile manifesto, we purposely didn't make black and white declarations that, you know, documentation is all bad. We made more of a, you know, we like this thing more than that thing. And because there is no right or wrong way to answer that question, but the team should be able to always look at, are we delivering the right amount of documentation? Is it too much? Because the way that I often tell people I like to do just enough upfront design, just enough documentation and even sometimes slightly less just to see, Oh, you needed more, sorry. I can, uh, I'll I'll, I'll add a little bit more. I wasn't

Speaker 3 (08:56):

Sure. Versus just blindly going to some level that might be too much that's again, the agile mindset requires you to constantly think about what are the actions or the downstream feedback from my action and includes documentation.

Speaker 4 (09:11):

This is a constant state of adaptation, or there's a little bit of that information release

Speaker 3 (09:17):

Agility and agile manifesto really is about set preferences, like where we want to go to eat tonight. Well, that's great. I might want pizza, but no one else does. Or actually someone comes in and has food allergies and we have to adapt and have to deal with that. And so it's about being able to be cognizant

of what the constraints are of the business, in my analogy, that's the group. But then all being able to get the best out of the people that are working there with the knowledge that we all really know what we're doing. So how can we get that better?

Speaker 4 (09:46):

So it sounds like agile is a state of mind. It's approach. It's about being very honest and authentic. Also

Speaker 3 (09:53):

I'd say agile is three main things. It's pragmatism reflection and empathy because these are the three things that you need to exhibit in others and yourself and support teams and order to create what we look for, which is high performing teams. And if you're not using pragmatism in everything you do two weeks sprints, should they really be two weeks? Do we really need to do this thing to get to over the line, reflecting in your retrospectives, not just on the work that you've done, the increment, but also on the processes that took you there and the things that stopped you getting further and finally on the retrospectives themselves, it gets very recursive, but you should be reflecting on the reflection to see if there's a better way in which we can learn because let's face it. There's only one thing that's truly irreplaceable in this world and that's time and not two weeks has gone and we're never going to get it back, but we've got two weeks again.

Speaker 3 (10:47):

So how can we make more out of it this time? And if we're not really thinking on that basis, then we're just delivering to get a paycheck. And that's not what the organization needs from us, frankly. It's not what our customers demand from us. For me, the one sentence that I'll often try to describe what agile is. It goes like this, do everything you can to reduce the gap in time from doing something to getting feedback. And that works recursively. It works at the code level with unit tests where you get rapid feedback works at the application level where you can do acceptance tests and see features working. And it works at the business level, driving the organizations, developers with expected outcomes for a feature, and I should be able to have a closed loop to measure it. So it's, it's a very recursive model that if you can only figure out how to reduce the gap in time between doing something and getting feedback that's Agile.

Speaker 3 (11:44):

John Kern is a coauthor of the agile manifesto and he's an agile transformation consultant at adaptive. Dan is the chief Technology officer at Weaver, which makes software for architects and the construction industry. You can read the whole agile manifesto@agilemanifesto.org. This is the tech barometer podcast. I'm Jason Lopez. We're a production of the online tech news site, the forecast. And you can find more stories@theforecastbynutanix.com. Thanks for listening.