

Business Analysis in an Agile Context Webinar Transcript

Adrian Reed:

Well, very good morning. Good afternoon. Good evening, depending on where you are tuning in from. My name is Adrian Reed and a very warm welcome to this BA community webinar.

I'm really pleased that you're able to join us and you're able to join us live. And in fact, if you saw the webinar marketing, you probably realize that you're actually quite lucky to be attending live because this webinar sold out. The topic was so popular that unfortunately we had to close registration early and that's no surprise because of course we're going to be discussing some very agile type topics today, which is always a popular and sometimes controversial topic.

I'm really pleased to be joined by Kent McDonald today. I asked Kent what he wanted me to say in his intro. And he said, "surprise me", which is always a really dangerous thing to say.

I've known Kent for - I was trying to work this out - it must be nearly a decade now. I think the first place we virtually met was on Laura Brandenburg's blog [Bridging the Gap](#). Kent you used to blog for Laura, right?

Kent McDonald:

I wrote a couple articles for her. Yeah.

Adrian Reed:

And I think we ended up commenting on each other's posts. Then I've seen you present at the [Building Business Capability conference](#). And, and as anyone who knows Kent will know, he writes fantastic [blog posts](#). If you're a subscriber to my mailing list Kent gets mentioned in them all the time. So do check out the blog there.

And of course, Kent has also written a couple of books: [Beyond Requirements](#). And... Oh, I'm having this all terrible mental block... it's [How to be an Agile Business Analyst](#). Is that right?

Kent McDonald:

That is correct yes.

Adrian Reed:

That would have been horrendously embarrassing...

[How to be an Agile Business Analyst] is an excellent book. I couldn't put it down. So, so do think about getting that.

So Kent, thank you so much for joining us today.

Kent McDonald:

Well, thanks for having me on. I really appreciate it.

Adrian Reed:

Fantastic.

And for everyone that's tuning in live we're also gonna try a bit of an experiment today.

So on a previous webinar, someone gave the excellent feedback that it will be useful to have some way of remembering the key points. And I was thinking about how to do that and I'm thinking, well, I could do like a wrap up afterwards, but that's quite difficult because it's facilitating. So we're going to try a bit of crowdsourcing today.

So just before we start, I want to show you something. So I'm going to share my screen very briefly. Okay. So what you're seeing now is a [Google Jam Board](#) and in just a moment, I'm going to put the URL in the chat and you'll be able to actually click on that URL and join the jam board.

(Editor's note: while you can no longer edit the Jam Board, you can certainly see what the people who attended the session live put up there)

And what I want you to do if you're tuning in live is when Kent says something interesting - which I'm sure we lots and lots of the time - or if there's a topic that particularly resonates with you add a sticky note.

So you might put something like that, you know, "agile, product management, et cetera, et cetera, et cetera." And then what will be really interesting is we can, at the end, after the session, we've got a bit of a record of the themes that you folks found interesting.

Now you may well find, you need to zoom in because there'll probably be so many so many post-its on the sticky notes that you'll need to zoom in.

So, what I'm going to do now is I'm just going to put, pop that in the chat. So you can actually play with that jam board whilst we are on air.

You can, of course also ask questions in the Q and A panel and I will be monitoring those questions as we go along.

So before we start Kent or as we start, probably one question that we should kick off with is ***what do we actually mean by agile? And is that the same thing as agility?***

Kent McDonald:

I'm glad we're starting with an easy question today.

Boy, there's a lot of different ways to go with this. And I think unfortunately agile has become something that tends to be highly dependent upon who you ask about it and what it is.

I like to think of it at the very core. And then I've kind of expanded it for what being an agile business analyst means.

But agile at its core is really about focusing on outcomes. So what outcome are you trying to get versus output. And as you're proceeding towards that outcome, using feedback to learn and revise your approach. And I think everything that has ever grown up around agile really comes back to those two things.

The whole agile versus agility split. I think a lot of that comes about because people were starting to see agile being taken by the tool vendors and process pitchers, which is kind of funny given where it started.

And so they've started saying agility to kind of try to differentiate it from that, but now they're starting to throw processes around agility as well.

So at the end of the day, I think it's important to remember that agile is kind of more describing how you approach things and it's how you approach getting towards the outcome. You're looking for using feedback and learning to get there.

And I like to keep it that simple, frankly. All of the other stuff can often be window dressing and can distract you from what you're really trying to accomplish.

Adrian Reed:

Yeah. So the outcomes and agile is the, the mechanism, the mechanism for achieving those outcomes, learning along the way. And, I guess then agility is the adjective, it's what you've become or it's what the organization becomes when it's able to respond to, you know, when it's able to be nimble and respond to change.

So how would you say Kent something like the agile manifesto fits into that question and that conversation?

Kent McDonald:

There's a little bit of history there. And then it's now almost, let's see it's oh boy, it's almost, I'm over, it's almost 20 years old.

And ultimately, it was a group of software developers that got together that said, you know, we've kind of got people treating us bad. So we want to put a status report of where we think things should be better and let's tackle some of these problems that software development teams often face.

And again, the key part of that whole [manifesto](#) - a lot of people focus on the four value statements that are in there - but the really important part is the very first sentence. It's "we're uncovering better ways to develop software by doing it and helping others do it".

So again, we're learning through experience and figuring these things out. That's the important part.

All the other stuff is kinda more just, for example, "here's some things we figured out, like maybe we should worry more about individuals and how they get together and get along with each other more so than processes and tools" and things like that.

What the agile manifesto did is it gave a rallying call at that time back in 2001 for people that says, "I can kind of correlate or I can kind of relate to how they're talking about working. We should try and work that way."

Adrian Reed:

Yeah.

Kent McDonald:

And then it's gone from there to be commercialized unfortunately and misunderstood.

Adrian Reed:

Yeah. Commercialized and almost, I mean, there's this in some cases, quite a lot of dogma about it. It's sort of received like it can't ever be questioned.

And, it's interesting that you mentioned the value statements. Those are the things that people focus on, but there's a [whole set of principles](#) in there, which I think are equally interesting, but less talked about. It's worth considering those as well.

So thinking about methods, cause you mentioned agile as being a way of achieving - or mechanism of achieving - the outcomes. And of course, there are a whole bunch of agile methods, probably one of the most common frameworks would be [Scrum](#) as an example. we might be using [Kanban](#) or one of the many other frameworks or methods.

But in your view, should we adapt a chosen..., Like if you're doing scrum, should you take scrum? I mean, let's face it. The [Scrum Guide](#) is what like 16 pages. ***Should we take Scrum as written and just implement it? Should we adapt it? What's your view?***

Kent McDonald:

The key thing to remember with all of the frameworks - Scrum in particular and the bigger brother of Scrum SAFe is also an example. Some of those frameworks date all the way back to when the manifesto was created, because each of the individuals that came to [that ski resort in Utah in the United States](#) had their own way of doing things. Those “own way of doing things” ended up becoming the variety of different frameworks that you saw. Although most of those frameworks went by the wayside because the folks that created Scrum are really good at marketing and they had certifications,

All of those frameworks are basically “this is a set of techniques and practices and ideas that worked really well in a certain circumstance, certain situation” which is great.

Where people can sometimes fall down is if they think that, “Oh, because it worked in that situation, it's also going to work in my situation.”

So the key thing with it is: there's nothing wrong with picking your framework to start out with, as long as you realize that the minute you understand how all those pieces in that framework fit together, you need to start tweaking it to fit better in your situation.

One of the main things I do with teams that I work with is to figure out, okay, well, when we're starting out - might be with a new team, it might be in a new company – I don't know much about the context. I try to find out as much about the context and the environment in the company as I can, but I know I'm not, I'm not going to know everything right up front.

I'm going to say, “we're going to pull this particular framework off the shelf - or a personal version of it, if you will - for what I've found to work. And we're going to start with this, but the only thing we're really going to make sure we do is have a mechanism to try something out, see how it works and adjust accordingly.”

Adrian Reed:

Yeah. And it's interesting, isn't it?

Because as you were saying that Kent, I was thinking that there are elements of any methodology that it makes sense to think about and change. There are probably some basic principles that you would always want to see that like the ability to see stuff and adapt to it. A showcase and retrospective or whatever they happen to be called in whatever methodology, unless there was a very specific reason that I can't really think of it. It would be unusual to take them out, for example.

Fantastic.

We have questions rolling in, which is excellent. We'll take our first question which is - and I hope I'm pronouncing your name right, apologies if I'm not - from Siddharth. And Siddharth asks ***how can a BA, be part of a technical, or micro service oriented, scrum where the product owner does not have any control on the product backlog?*** Ooh, product owner not having any control on the product backlog... that's quite concerning.

Kent McDonald:

In that particular situation - I haven't been in that exact situation before. But I was working with a company once that had a very large product that they were trying to support. And they had organized

their teams into...there were certain teams that were working on directly customer facing functionality. And then there were some other teams that were working kind of more on the overall platform set of it.

So basically there, the people they were serving were the other teams that were actually providing customer facing functionality. And what happened with those particular teams is that they did still have someone doing product ownership. But those people ended up actually being... Their thought was: let's actually have the architects for the overall product be the people that are doing the product ownership for the platform teams. Because they basically need to make the decisions about what are the particular aspects of our overall platform service that we need to provide to these other teams in order to make sure we're supporting what they're trying to do.

So that's kinda how it worked there. And that's a way of making sure that someone was responsible for doing product ownership. Basically again, focusing on outcome, making sure that there was a shared understanding, making sure decisions were getting made. We're really looking at it from the standpoint of what technical capabilities do we need to provide.

Now, in those particular teams, they did not have business analysts, but there may have been need for some form of business analysis to understand how does the needs of our final customers translate into what kind of technical underpinnings do we need to provide?

So if you are in that situation - I'm guessing you are, since you asked that question - I would look to see: 1) is there any - it may not be a direct tie -but is there any indirect tie to what our ultimate customers are looking for?

And 2) is there some need to get some understanding about what outcomes are they looking for? What are they trying to accomplish at the end of the day with that product? And how does that translate into what we need to do from a services standpoint if you can't see that direct tie.

And it's very separate what that microservices team is doing compared to what the ultimate customers are looking for.

It might be that you need to start saying, Hey, maybe I can be better used on a different team. Maybe there's another team where my skillset and my viewpoint and really being able to understand business and technology might be better suited in a different place. Cause sometimes there's going to be places where the unique skill sets of business analysts aren't quite as necessary, unless you've also got a lot of good technical chops where you can translate that into say, okay, so that's what we're ultimately trying to do. This is what those customer facing functionalities need to do, but they also need this underlying core architecture piece that we can also do as well.

Adrian Reed:

Again, it's interesting. That's a really nice theme that's emerging, I think, here around outcomes and seeing the link to the outcome. And if that isn't there, then perhaps it's not a BA role or perhaps it's a different type of specialist BA role.

Our next question comes from Simone. Simone asks ***"I keep hearing Kanban referred to as an agile methodology. I'm confused as I thought it was a planning tool from lean six Sigma within my organization. It's sometimes used for planning within scrum. Please. Can you clarify? Thanks."***

Now if I'm, if I'm right. I think Kanban originated from the Toyota production system and lean, but that might be my misunderstanding.

Kent McDonald:

I believe it actually originated at a grocery store in Japan.

Adrian Reed

Oh, really? Wow.

Kent McDonald:

From which the folks at Toyota borrowed it and started using it in their plants. But so it is pulled from lean manufacturing, but it is something that, that software development teams can use.

The way I like to think of it is: If you think of Scrum, basically what you're doing there is your limiting your work in progress by saying, here's what we're going to work on in the next two weeks or whatever length your sprint is.

What Kanban does is a couple things. One, it says, you know, we're going to even limit work in progress down further. So instead of blocking off and saying, we've got a chunk of work that we're going to do in the next two weeks, we're really going to focus on just making sure pieces of work flow through the system as quickly as possible.

The teams that I work with we'll often mix the two together. We don't give it any name - although some people like to call that Scrumban or Kanscrum or whatever - but we don't. We don't call it anything other than "this is how we do stuff".

Where we might kind of have an initial discussion to say, what's the next set of stuff that we need to be working on. And we still keep a backlog, but we then have the flexibility to say, "okay, as of right now, what's the next highest and most important thing that we need to get through the system?" and then just kind of progress it through our process. Not only the backlog refinement process, but then also the development process as well.

We do it that way because it gives a bit more flexibility to help us understand, to adjust and react. When we learn things, we can say, "Oh, now that we know this, this other thing might actually be better to deliver next rather than this thing that we thought we were going to deliver".

Adrian Reed:

Yeah. It's, it's really interesting.

And that limiting work in progress... In a completely different context, whenever I feel overwhelmed - like when you're on a project and you feel overwhelmed - one of the things I like to do, and I actually will sometimes do this physically with sticky notes. What am I actually trying to balance with? What work do I have in progress right now? The reason I feel overwhelmed is cause I'm trying to do too much simultaneously. And we all know that multitasking is a myth. Right?

Kent McDonald:

Exactly.

Adrian Reed:

Well, let's stop doing these things. Let's finish this and then pull that from the backlog. I guess the idea of pull becomes important in Kanban as well.

Our next question is from Michelle. Michelle asks, "***what are your views on the Gherkin syntax?***"

For anyone that's not familiar with Gherkin, that's the [given when then](#) type acceptance criteria syntax.

Kent McDonald:

That could be a potentially loaded question...

I think it can be helpful. I generally refer to that overall set of way of describing things and I call them [examples](#).

I have run across some teams that feel the need to require people to put everything into a Gherkin format, even going as far as expressing their backlog items in terms of Gherkin.

I don't go that far.

I basically find it [Gherkin] can be helpful to explain fairly complicated complex business logic. So I'll certainly pull that out and use it in those particular cases, especially if it's some sort of procedural thing that something happens and then something happens as a result of that. Gherkin is great for that.

But if I'm working on say a business rule that has several different inputs to it and certain outputs, I'm gonna use a tabular format.

The main thing with all of these examples is not: "you always should use X particular format." It's basically: "here's a selection of things we could use. Let's figure out what works best for our team to use. And then we'll do that."

With the team I work with right now, the way we roll is that if there is some complex business logic that I need to convey, I will switch over to using that format. But don't always do it.

Sometimes it's a judgment thing to say when should we use it, When should we not?

Now I should also note that the team does do automated acceptance testing and uses a tool called [SpecFlow](#) that basically requires things to be in that structure [Given-When-Then]. So the developers are certainly building stuff out in that, whether or not I am actually laying that out ahead of time.

My title is delivery lead, but I'm basically kind of doing a product owner/BA type role. It varies depending on the nature of the backlog item.

Adrian Reed:

Yeah. I think that's such an important point Kent. Actually, I think it's probably fair to say there's just no one right answer to just about anything in agile or projects or probably life.

Given-When-Then has - as you say - has its place when there's that logic, that procedural stuff. But again, I don't know if you've found this, but I find when you get lots of ands in there like, it becomes really difficult. I take the view that if the product owner can't read it, then what's the point? If a product owner can't genuinely tell whether that [logic is correct] then maybe some format of diagram is better, you know, a picture, a flow or something.

Kent McDonald:

I would hope that the product owner would be able to create those Frankly. So if they're not able to even kind of think in that, then you probably got to use something else.

And while we're on the topic of formats real quick, I will note that I have the same outlook on the [good old Connextra format](#) for user stories: As a - I want - So that. It's great training wheels. It's great in some situations, but if you were to look at a backlog that I've been working on, you're not going to see it that often. And it's because there's other ways of conveying that context and some cases that makes perfect sense to do it that way.

In other cases that's overkill.

Adrian Reed:

Yeah, totally... Fantastic.

So, before we move on and take some more questions - and there are lots of questions coming. Please, please do, keep them coming in.

I see we've got we've got 30 people looking at the Jam Board with stuff being added to it. So that's interesting. We'll kind of bring that in at halftime and see what people are writing in there as well.

But I just want to briefly change tact a little bit and talk about product management versus project management. I'm going to put my hands up here and say my background was originally financial services. I lived in a project management world as a business analyst and it was quite a shift for me. The two disciplines fight in some organizations, it seems that there's this there's this tension between them.

What would you say the differences between product management, project management and can they coexist?

Kent McDonald:

So I'm gonna go back to a familiar trope here that I've been saying a lot already. I think one of the differences between project management and product management is their focus.

Project management tends to focus on output. How are we going to get this thing done? They're kind of expected to be handed what it is they're trying to accomplish. They're expecting to be handed the constraints in which they're working. So that becomes scope time and cost, right? And then they're going to make sure that they get it done within those constraints

Very valuable, important efforts. But it gets saddled with certain things.

Like it's the assumption that we've got a team of - we've got a group of - people that are pulled together to work on a specific thing. And when they're done that team - that group; and I'm, I'm correcting myself there for good reason - that group of people will get split up and often do other things.

I also grew up in financial services and the project world, and one of the biggest banes to my existence was the fact that for a while, people didn't realize that people couldn't multitask. And so there was the whole idea of everybody would be much more efficient if we had them working on seven projects at the same time with seven different groups of people.

Adrian Reed:

Yeah... Because it's just really bad if anyone has any downtime for things like thinking or sleeping.

Kent McDonald:

We couldn't and possibly let them not be doing something all the time!

So that's, that's project management.

Product management, however, is really focused more so on understanding what that outcome is and they are more worried about the problem they're trying to solve.

So, in some organizations you see those two functions fight with each other and it's usually because there's not a clear understanding of what each role is supposed to do.

You can have organizations where those two functions do work well together because the product management function is basically saying, here's what we're trying to accomplish. The project management function - which a lot of times now is known as delivery managers or release managers - are then taking it up from there and responsible for making it happen, especially in large organizations

where there's a lot of coordination necessary In those situations that works great. If you've got a small team, you probably don't need both.

Adrian Reed:

Yeah. And just on that Kent just very briefly before we move on and take some more questions... Another question comes up quite a lot is the difference between a product owner and a product manager. I know that's a whole can of worms, but what's your view on that Kent?

Kent McDonald:

Let's talk about agile and agility again...

No. So the thing that I have found most entertaining about the product management/product ownership debate is always trying to guess what community someone comes from when they're talking about it, and you can tell based on their position.

Scrum originally did not have a product owner way back way back in the very beginning. They, referred to them as product managers.

Somewhere early on Ken Schwaber - I think it was Ken - said, "You know what? We need to differentiate our product person from those mean product managers out there that weren't working very well with the teams. So we're going to call them product owners."

And the way he positioned it was is that this was actually a person that owned actual decision responsibility and could make those decisions. But at the same time could spend a lot of time with the team. But [except for] the very smallest organizations that is kind of like a unicorn.

Because if you want the person that is able to make all the decisions that has all the background knowledge you probably aren't going to be able to find that person. But then [also] can spend an inordinate amount of time working with the team - which don't get me wrong, that's great. If you can get it, it's very hard to find someone that has all of those qualities.

So if you've got people that are saying, "Oh yeah, yeah, product owners do all those product management things too" generally, they're coming from the very hardcore Scrum community because they believe that "our term is correct."

When you've got people from the product management side of the fence they'll say, "yeah, but there's all these things that product managers do when they're working on a for sale external facing product. They've got to figure out pricing, they got to figure out marketing. They got to understand what the problem is. Oh yeah. And they got to work with the team."

What I usually see [happening in real life](#) is that you have someone doing **product management** that's trying to understand what problem you're trying to solve. They're keeping an eye on the outside on the market. They're keeping an understanding of what their customers are trying to do.

You've got **product ownership**, which is primarily inward focused working with the development team, making sure there's that shared understanding and making sure the development team has what they need.

And then you have **business analysts** fitting in there that are really understanding what are the business rules we've got to worry about? What's the data we got to worry about? What are the processes we're supporting?

Now, it's possible that one product person might do all three of those things, or those functions might get split up depending on how complex your organization is, how complex your product is and all those things.

So I see product ownership is actually a subset of product management.

All the Scrum people out there are now probably rolling over because I just said that.

Adrian Reed:

It's a controversial topic. So I just want to bring in a comment that Um I've sort of moved my screen that Tony has made, which is can a business analyst be a product owner on a project team? And and I mean, you've sort of addressed that by saying yes, they could.

And it's interesting. Cause I was speaking in the last webinar I did with Alan Kelly. Do you know Alan Kent?

Kent McDonald:

I do. Yeah.

Adrian Reed:

So I was having a conversation offline with Alan we were playing with ideas. Because IIBA's definition of business analyst is anyone who does any business analysis. That is an argument that says all product owners are business owners. I think that they do elements of business analysis. There's an element of a business analyst role in the product owner role. So that's probably equally controversial amongst some quarters.

So fantastic.

From same questioner: ***for a new business analyst who is the only one who is interested in the agile methodology in my organization, how do I introduce this and make this the culture of my organization?***

So that's again from Tony.

That's a huge question. You could spend the rest of the webinar answering it. But any tips?

Kent McDonald:

So that was me in 2004.

I was walking around a Barnes and Noble, which for the folks in the UK, I think it's probably the equivalent of Waterstones. Is that right Adrian?

At the time, Barnes and Noble had discount book tables. I walked past there and I saw this book called [Agile Modeling](#) by Scott Ambler.

And I think Scott has heard me tell this story before, and I don't know what his feelings are about me mentioning that his book was on the discount table.

But anyway...

I looked at that, I picked it up. I said, "wow, this sounds a lot like what I do."

So, at the time I was working in a data warehousing group for a health insurance company - another probably odd type of organization for people in the UK. You have some form of private health insurance, correct?

Adrian Reed:

Bupa is the most common one.

Kent McDonald:

Probably not nearly as messed up as ours is...

Anyway, I was working in a data warehousing group, which at the time were infamously not interested in anything that's iterative or incremental or agile or anywhere close to it.

And I said, "but this really resonates with me. It's really like how I like to work."

So I started introducing some of the techniques that seemed like they would be helpful in various particular situations. And then one mistake I made is to actually announce that this was this thing called "agile."

So what I learned from that is you do it in a guerrilla fashion where you're kind of just finding techniques that would work in particular situations and just try them out. Don't say you're doing agile. Don't say you're doing scrum. Don't say you're doing any of those things.

Just say, "Hey, you know, we've got this particular problem we're facing, why don't we try this out and see how it works?"

And then if it goes well and people ask you afterwards, "well, how'd you do that?" Then you say, "well, let me tell you, there's this technique and this approach we used, and this is why I picked it. And here's why it worked in this particular situation."

You're going to have a lot more success I've found if you demonstrate how it can work, not be dogmatic about it and just say, "well, in this particular problem, this is how we did it. And this is the result we got." People will usually say, "Oh, I want me some of that. Let's get some of that going here."

Adrian Reed:

Yeah. It's that old adage of credibility through delivery. Actually show rather than tell.

Okay, let's have a bit of fun. Let's see what people are typing on the Jam Board. Cause this was a complete experiment.

And just to add to the jeopardy, because it's on my other screen, I can't actually read it. So we'll be revealing live what the comments are...

Kent McDonald:

Hopefully you're fast.

Adrian Reed:

Okay. So people have got a user stories in the age of microservices, making stories, outcomes, centric stories.

This is how we work inside out, which is another key point you made. Rather than labels, we've got adapting a framework, tailoring.

We've got some stuff about product owner and product marketing.

Oh, here's a question someone's typed up. ***What do you think about PMIs approach to disciplined agile? How do you see BA in such a diverse way of working? So maybe rather than just focusing on PMI's approach, what are your views on disciplined and scaled agile more generally?***

Kent McDonald:

Well the dirty little secret that PMI may or may not want you to know about their approach to Disciplined Agile is it's actually [Scott Ambler's approach to Disciplined Agile](#). They bought it and then they hired him on to help with it.

I have a lot of respect for Scott. One of the things that he's been doing a long time has really been trying to incorporate a lot of the entire life cycle if you will, of, of developing software.

He's also real big into thinking about architecture and thinking about databases.

And he was talking about agile analysis very early on, but he wasn't necessarily as complimentary about business analysts at the time. And at the time where he was talking about it, it was one of those "you definitely need people doing analysis. And there are certainly people with those specific skill sets, but you don't necessarily want to have people that that's all they do."

You want to make sure that they're understanding and the breadth of what they do from an analysis standpoint is really focusing on the analysis as opposed to saying, "Oh, no, all I do is document requirements." You don't want that.

The way I describe it now is it's really one of the main things that BA's should be wanting to do is to make sure they're building a shared understanding about what the problem is they're trying to solve. And all of those analysis techniques are great ways to help you do that.

Adrian Reed:

Yeah.... Am I right in thinking – my business change history might be failing there, but he was involved with RUP wasn't he? The Rational Unified Process. There was the hump diagram. Anyway, this is none of this is relevant for this chat, but it's sparking your own.

Kent McDonald:

He worked for IBM for a while and was with Agile Unified Process and then went off and did the disciplined agile. And then PMI says "we got to get us some agile." And so they brought him on...

Adrian Reed:

Yeah, small world.

Okay. So let's take our next question from anonymous, an anonymous attendee.

And of course, if you do want to ask a question anonymously, that is completely fine.

The anonymous attendee says that they are "a brand new BA no formal qualifications headhunted from a brand new mental health NHS national health service system to work as a junior business analyst on the expansion of the service."

That sounds fascinating.

"I would like to take some courses in the specific skills needed for a BA specifically, it looks like in an agile context cause they put in brackets, agile mapping lean and six Sigma people management user experience rather than a full BA qualification. ***Any suggestions on the most important skills to develop or get a qualification for, or indeed any short courses worth mentioning?***

That's a really thoughtful question about skills rather than just the certification element.

Kent McDonald:

Yeah, I like that line of thinking.

Boy. The most important ones.

So if you think back to my defining characteristics of an agile business analyst:

- outcome over output
- building shared understanding
- using feedback to learn
- making sure decisions get made
- thinking about context.

Those are the five main things that I focus on. The skills that are important in there. All of them have a very large component of communication to them.

Adrian Reed:

Yes.

Kent McDonald:

And with so the team that I work with right now, we for the longest time were co located. We had a lot of advantages while working in the same room and everything. When March came around and the pandemic, we had to go remote. Which personally I'm a huge fan of...

Not a huge fan of the pandemic, but I am a huge fan of working remotely. Just wanted to make sure I stated that correctly.

But what I found is that your communication skills become very important and it's more than just the verbal communications skills at that point. Because it's one of those principles you mentioned earlier was the best form of communication is face to face.

So certainly there's value in being able to be on a video chat and everything. But I do also think there's some value in balancing that against using the right mechanism at the right time.

And I've been influenced on this [by the folks at Basecamp](#) - used to be called 37 signals. And they do a lot of things by writing because they basically say there's certain types of purposes of communicating where it's actually probably better to write it out.

One because it might be just you're conveying information to a lot of people it's not something that requires a lot of back and forth. Plus, if you write it out, it forces you to kind of think through it a little bit more involved.

So I would say one of the most important skills is understanding what are the different types of communication channels that you need to use and knowing when to use which one.

Adrian Reed:

Yeah. I mean, it's interesting, isn't it?

So much of what you're saying there, Kent resonates with me because this is analysis. I mean yes, of course the toolkit is important. The models are important, but fundamentally, I mean like all change, however we, however much we dress up, it's a human endeavor, right?

It doesn't matter how good the tech is. It doesn't matter how good the processes are. If the people aren't on board, if they don't understand, it's not going to work and, and actually how do humans collaborate or communication.

So that really highlights those sort of communication and shared understanding skills.

We'll move on to our next question from John. John says that he's a PM who migrated to being a BA under, under a waterfall with sprints and stories environment in an agency. Now they're looking for a new role, preferably client side or public sector moving away from marketing. ***His main barrier to employment is agile experience. Any suggestions on how to get around that?***

Kent McDonald:

Interesting...

I've had this question before.

And again, I'm glad he's asking about how to get experience rather than "will certification get me there?"

There's a couple of different ways you can do it. One, if you're in an organization now, see if there's any opportunities there to - if nothing else - insert those practices, guerrilla style.

And then when you're talking to the people that you're interviewing with, don't necessarily just go out and say, "Hey, I've done agile" because everybody does that.

I've gotten to the point now where I can tell where people haven't really had, what I would refer to as, as an actual experience working in an iterative fashion and using feedback to learn because they throw all the buzzwords out there, but they really can't describe it.

But if you're able to actually talk through what you did, how that helped you and your team to get outcomes and how you were able to cycle through things and learn from it, that's very powerful.

If you don't have the opportunity to do it in, in a job that you're in right now, for, for whatever reason, there's often ways of applying [agile techniques]. Even if you're doing things from a volunteer perspective, or even if you're just doing things as a side gig.

So it might be that, cause I know some people that say, "how do I get some experience being a little bit more hands on with things?" I often encourage people to start a website, start a blog or anything. And even if you just use WordPress, the function of going through and figuring, tweaking the settings and everything like that, you can use that as a platform to learn how to try something, experiment with it, see if it works, tweak it, change it.

If you're able to explain that and why that helped you and how you would apply that to the situation of building a product or an internal application, I think that will go a long way with interviewers.

Adrian Reed:

Yeah. And Kent just to echo what you're saying there, I think from a volunteer perspective, in an organization, I know you are, and a lot of people on the line will be familiar with IIBA (the international Institute for Business Analysis.)

And you know, IIBA has local chapters and local chapters are normally really calling out for volunteers and most like talking about websites. Most local chapters have websites that need enhancing and maintaining.

If you're a volunteer looking to get experience and you go to your local IIBA chapter - and there's one here in the UK. If you happen to be in the UK and said "I want to help you manage your backlog for all of your, requests for your website." The answer will almost certainly be yes, because who's going to say, no? Volunteering is always a good option as well.

Okay. So our next question comes from Michelle. Michelle asks, "how inclusive do you think agile or specifically the scrum method is? ***How can we ensure that introverts or reflectors are not***

disadvantaged by all the ceremonies? That's an interesting question. I've never really thought of that before.

Kent McDonald:

So if you're speaking directly about introversion I'm an introvert.

Adrian Reed:

Yeah, me too, actually.

Kent McDonald:

So as long as you have facilitators on your team that are worth their salt - that really should be a scrum master in the case of scrum. They should be looking for ways to encourage everybody to be engaged at the level of which they're comfortable.

If you are in a position where you are being that coach or that facilitator, one of the key things you need to make sure you understand is how everybody is comfortable.

So I find myself in that role quite a bit, and it's a matter of getting to understand your team members' preferences and to know kind of the tells they have. Most introverts have some way of subconsciously conveying that they're pondering or thinking of something. These types of tells initially are best trying to pick it up when you're able to see them in person.

But I found that with the team I work with now, I've been able to pick up even since we are not in the same room and we don't generally hang out on video all day, but we do have an audio channel on all day. I've still been able to pick up when people are considering things or I will actually reach out and say, "Hey, just checking Adrian, what do you think about that" if they haven't spoken up in awhile.

But I do think that when facilitated properly all the techniques can work very well with extroverts and introverts. The thing with that is though is that you also have to balance not doing the ceremonies for ceremonies sake.

So I try to keep official meetings to a bare minimum.

Even though we're a collaborative team because and I I'm fully aware and being a writer probably has helped me learn this. There's just some time where people need to get off by themselves so they can get into flow. So I'm very conscientious about trying to make sure we're not doing things that disrupt people's ability to get into the, get into flow.

Adrian Reed:

Yeah. So important that kind of deep focused work. I mean, that's the real hard stuff, isn't it?

And as you were talking there Kent, I was thinking that actually this new distributed way of working or, or where everyone's distributed rather than, you know, typically have some people in the office, some people remote, some people in another location, it's kind of a leveler, isn't it?

As you were talking, I was reflecting and mulling because I'm an introvert. People never believe me when I say that I really am. So I find presenting exhausting, for example.

But I found things like the nonverbal feedback that some tools like Zoom has a nonverbal feedback where you can give a green tick or a red cross. And when I've been facilitating, if I just want to do a quick temperature check if everyone's like, okay, And we can move on. Can everyone give me a green tick? And if you think we need to stop, give me a little coffee symbol type of thing.

In a real-world environment, I've laminated cards with stop, go faster.

And I also have given to me by a friend in, well, I can't really whether it's from Belgium or the Netherlands, but I met him in Belgium, an Elmo doll, like the physical Elmo which is great.

Because it's like, hold up the Elmo when it's enough, let's move on type of thing.

So you can incorporate playful stuff like that into your practice as well.

So our next question - I think it's observation actually - comes from Tim. Tim says "project management deliver what's been contracted or asked for within time cost and quality constraints, a product owner or BA or systems engineer discover what's needed, build the right thing and build it right. What's needed is often different to what's contracted or asked for. "

That's quite nice, a nice way of summarizing it. I think there

Kent McDonald:

Our work here is done!

Adrian Reed:

Let's write that one down.

Okay. So then Jason asks *what's Kent's view on the coexistence of agile waterfall and DevOps methodologies in organizations and how we as BAs can effectively challenge the associated tensions, which often manifest as a result.*

Kent McDonald:

So I'm going to start off on this one that I feel like sometimes I need to double check my understanding of Dev Ops because I've heard oftentimes where it's been contrasted with agile, and I've always just wondered why that was the case. Cause it's actually kind of just..., It's embodying a lot of the same ideas as agile does.

[DevOps is] basically just focusing on the fact that for the longest time, the people that developed products and the people that were responsible for operating and releasing the products weren't necessarily the same folks and they didn't really get along very well together. So there was this need for DevOps idea to say, "Hey everybody, you should all work better together." And it's, it's actually something that fits in nice. It's pretty much bringing a technical perspective back to agile software development is really what it's doing.

In fact, when I was the chair of Agile 2013 we made a concerted effort to bring in more DevOps related content into the conference at that point in time because it was just starting to happen and get new. So those two definitely fit well together.

When you have an organization that wants to do a mixture of waterfall...which by the way was never called "Waterfall" before agile came along, go back and check. It was basically because the agile folks needed some kind of derisive term to refer to all this stuff "that's not agile."

Adrian Reed:

It's a bit of a straw man.

Kent McDonald:

It's a huge strong man.

Anyway, if you've got an organization that's doing software development work and they're saying, "Oh, well, we need to decide if this particular project is waterfall or agile." I usually view that as a sign that the organization doesn't quite understand - it doesn't share my understanding of what agile is.

So they're still thinking of agile as a methodology, as opposed to if you're acting in an agile manner.

You're going to say, "okay, what's the context that we're trying to do? What's a good way for us to, to try things out, learn from it and adjust and you know, what are the constraints we're facing?" And we might find that there are certain aspects to doing a phased delivery that might make sense. I've also found that one of the key underpinning key underpinning assumptions of agile [I meant to say waterfall here] is that you can figure out everything you need to know at that very beginning.

And then once you've done that, nothing will change after that.

In software development, especially, I don't think that's the case. I think you always are learning more. Even building a house. Construction is one of the places where software development got that whole phase lifecycle from. Even though the methods are pretty well understood, and it's been done millions of times, there's still some things you can learn.

So if you've got an organization having this whole big process, to decide, do we want to do agile or waterfall? The folks that are trying to push making those decisions, I don't know that they really understand what you can get from running things in an agile manner.

Adrian Reed:

There's a lot of buzzwords sometimes isn't there? Sometimes it's about it's about the perception rather than the reality of what the approach is involved.

So I'm going to bring in a question from Richard here, because we were talking about learning. Richard asks, "***Kent talked about the importance of learning in agile and adjusting your approach in the light of that learning. In relation to the product and what it does, effective learning can take time in my view and experience people's or organization's first reaction to something is not always how they feel about it after using it for awhile, but it seems very true. Any thoughts about this and whether, and how to include this time in an agile project?***"

And it is true. I can't remember who said it, but I remember hearing someone once say that nobody ever builds thinking time into a project plan, which is an interesting observation.

So any thoughts about that, building in the ability for people to see something play with it, form a view and then give feedback?

Kent McDonald:

So I've been working on this particular product for a year now and it's, it's for an internal audience of two people. I mean, there's a lot of complex business stuff involved, so that's the key part of it.

But we found is that you can do demos every two weeks until you're blue in the face, but you really don't get meaningful feedback until they actually start using it in their day to day jobs.

The best thing you can do is to figure out how can we get them something fairly quickly and get that to them in order that they can start using it in real life because a lot of users are gonna be, "yeah, yeah, yeah, I'll try it out." But they don't really tell you how it's working for them until they have to use it and it's impinging on their daily work.

So how can you start getting them to use that as soon as possible to give yourself some time to make revisions and corrections in some cases to what you've delivered based on how they're actually using it, unless you can figure out some way to get meaningful, informed feedback from them using it

beforehand. The best bet is to say, "let's get them something as quickly as possible." Give yourself some time before it really has to be in place and then put a good mechanism in place to get that feedback.

Adrian Reed:

Definitely.

Okay. So we're nearly up against time.

And Kent you kindly agreed before the session to take a copy of all the questions that have been asked and to write a blog post about all the questions. So what I'll do is there'll be a wrap up email that will go out to everyone. I'll put the address of your blogs so people can keep an eye on that for when that goes live.

But just before we looked to round off, Kent, if there was one takeaway, one soundbite, one technique, one thing you think you'd want to want people to take away from this webinar, what would it be?

Kent McDonald:

I'm going to go with there is no such thing as best practice. And the reason for that is, is that best practice implies there's one thing that you can do in a specific way that will work in every situation.

And apart from trying something out, learning from it and adjusting, I have not found any other best practices.

So it's really important to understand the context in which you're working and don't be afraid to adjust your approach based on what you learned.

Adrian Reed:

Yeah. Context is everything.

I remember hearing. I think it was Simon Wardley - this has always resonated with me - say "by definition, best practice is always past practice."

And ever since I've heard that, I've thought you're right. I mean, it's interesting, isn't it?

So, fantastic.

I'm sure Kent, we could go on talking for another hour and then another hour, but our time is sadly coming to an end.

So thank you so much for taking the time out to be here. I am absolutely certain that people will want to stay in touch with you.

Kent, I want to know about how to get hold of your books. How can people stay in touch? How can they connect with you?

Kent McDonald:

Well, first of all, Adrian thanks for having me on. This is a great. I appreciate everybody that's over in the UK and Europe that are joining in the evening. So I know that's a little taking a little bit out of your personal time.

Best way to keep up with me is to go to my [website, which is kbp.media](http://www.kbp.media) where I publish resources for product people. Mostly focused on product management, product ownership and business analysis. I do come from a business analysis background, so all, all of it tends to have a BA t slant to it.

You can find me on [LinkedIn](https://www.linkedin.com/in/kentmcdonald/) and those are usually the best two places right now. I do [a weekly newsletter](#). So sign up for that and hope you find some useful stuff out of there.

Adrian Reed:

Yeah, absolutely.

I definitely recommend get get a [copy of Kent's books](#) they're available on Amazon. You can get them in a Kindle version, if you like reading on Kindle, like I do. But just make sure you allow an afternoon because I found I couldn't put the thing down and always annotating it and so on.

So fantastic. Well, Kent, thank you so much for, for taking the time to, to be here and thank you to everyone who has tuned in live the, the Google jam board, by the way that we've been playing with.

I'll change it to read only after this session. So you can take a copy, you can take a look at it and you can download it. So that was just a bit of a an experiment.

The session is being recorded, so you'll be able to watch it back. And any of your colleagues that couldn't make it, you can of course, forward the link onto that.

But you will get an, a short email, it'll have a link to the recording. I'll also put a link to Kent's blog and LinkedIn profile, so you can connect with him that way.

There'll also be a very short survey. Now, the survey has just about three questions on it. And one of the questions is what topics would you like to see on future webinars? And one of the main reasons we've had a couple of agile webinars recently is because people were overwhelmingly saying agile. So please, please do put topics that you want to see those, you know, we'll, we'll add them to the backlog. We'll try and make it, make it happen equally.

If you're watching this thinking, I'd really like to present, I've got a story to tell. I've got an interesting case study, get in touch. You know, you don't need to be a "professional speaker" in inverted commas. It really is just a chat.

So thank you very much once again for joining. Thanks everyone. Enjoy the rest of your day and speak to you again soon.

Cheers. Bye.