[04:32] RESEARCHER:

Hi PARTICIPANT 9, how are you?

[04:35] PARTICIPANT 9:

I'm good. How are you, Researcher? Thank you so much for accommodating.

[04:38] RESEARCHER:

No, it's okay. I understand. I'm doing very well.

How about if we start with some introductions and we can go ahead with the questions?

[04:49] PARTICIPANT 9:

Sure. Absolutely.

[04:50] RESEARCHER:

I'll start with introducing myself, telling you what I do. And we'll go from there.

[04:59] PARTICIPANT 9:

Sure.

[05:00] RESEARCHER:

My name is [Deleted to preserve the participant anonymity] I'm from the [Deleted to preserve the participant anonymity]. I do research in software quality. I'm interested to understand how various software methodologies and software processes achieve quality. Currently, I'm focusing on Agile. I'm doing my research on Agile, trying to understand how Agile teams achieve software qualities in their development. I mentioned that in my messages, your participation is anonymous and your name wouldn't be used in the research and the name of the companies you work for or the companies you mention in your answers to the questions wouldn't be mentioned or used anywhere in the research. So, what we understood is your perspective and your view and your experience in practice. That's what we are interested in. Do you have any questions for me before we start?

[06:08] PARTICIPANT 9:

So just one quick question. Are we going to record this session?

[06:13] RESEARCHER:

Yes. Are you okay with it?

[06:17] PARTICIPANT 9:

Yes, absolutely. It's fine with me.

[06:21] RESEARCHER:

What's going to happen, we're going to transcribe the session and I will send you the transcript for validity and approval. Once you validated and you're happy with it and you feel free to add or you can feel free also to withdraw something from the transcript. And we can do that after we transcribe the recording.

[06:46] PARTICIPANT 9:

Sure. Yeah that's fine.

[06:51] RESEARCHER:

OK, how about if we start about introducing yourself and tell us what you do and talk to us a little bit about your experience.

[07:00] PARTICIPANT 9:

Sure. So, I'm basically having around 11 years of work experience in software development, business analyst and product management. So, I've worked in different geographies like from Hong Kong, Germany, Poland, and I work for different clients in different domains. So, I started my career as a Java developer when I was using Agile basically as a developer. And then I moved on to become a business analyst where I was gathering requirements from business and sharing it with the development team. So, then I did my postgraduation in marketing and then I moved on to product consulting where I was doing product consulting for e-commerce companies, for e-commerce compliance, for telecom clients in Europe and Asia Pacific. So, post that, I started my own venture as well. It was a social network that I started but couldn't make it work so I had to shut it down and move forward. So, then I worked with a company called Snapdeal, which is one of the top three e-commerce companies in India with around five million daily traffic users. A very Agile company with releases happening every two weeks and these were all in our goals to delivery. So, I was taking care of products before moving onto [Amway] where [Amway] I was leading European track for [Amway], all their e-commerce portals and the mobile applications for 26 countries in Europe. I was leading that and the requirements gathering. For the past one year, I'm working with [Deleted to preserve the participant anonymity] It's a Singapore-based start-up, which started its operations in India last year. I'm Vice President of product management here leading a team of managers and designers. So that has been my experience to date.

[09:15] RESEARCHER:

Okay, fantastic. The first question that's comes in mind is how Agile implementation is different across these companies?

[09:26] PARTICIPANT 9:

Sure. So, if you start with a company like [Deleted to preserve the participant anonymity] which is a capital organization, Agile was not too popular at that time. We started using it way back in 2008.

[09:42] RESEARCHER:

Correct yeah.

[09:44] PARTICIPANT 9:

So we were learning how we are going to create or define the velocity, how we are going to do a better measurement of story points. And how better it is from Kanban and from waterfall methods. So that was more exploratory of Agile methodology that we were already using at that point of time. Still nice for us. But then the learning was really good because we had teams who were working from US and there were offline teams working from India. OK. So how we were doing it, there was a guy that was a scrum master who was sitting in Boston and who was guiding the teams in India. So, we were basically understanding it from the guy that how it needs to be followed. So, it was more like a learning thing and then implementing this thing. It was very different when it comes through a company like Emphasis, which is a big company in terms of the technology side. And they have like huge, huge manpower, and resources.

[10:52] PARTICIPANT 9:

So, they had around one hundred thousand people in the company and Agile was practiced in some of the units, including mine. So, in this case, the Agile was mostly in the sense that when the requirement gathering was limited to defining only the MVP. So, the business or the clients, we're only giving this scope minimum by new product.

[11:17] RESEARCHER:

Correct.

[11:18] PARTICIPANT 9:

But once we have that finalized, then we used to come back with the new scope. So, the vision was not there, but only we were looking from the mission perspective. So, if the mission gets successful and the client is ready to pay more, then the Agile would be more and we have to look at it from the other side of the road map. Coming to a company like

[Deleted to preserve the participant anonymity] it's a full e-commerce company with a huge amount of traffic on a daily basis, a lot of new features that need to be launched. So that's where we actually started using the mature Agile, where you have to come up with a lot of new things and it needs to be delivered in a very short span of time. But if look at it from the journey, we had had six weeks off to two weeks of sprints in Snapdeal. So, the transformation happened from there. So, we can have a more refined product backlog, more quality checks and fast delivery as well. So that's how we evolved over the period of time.

[12:24] RESEARCHER:

Fantastic. Let's start with this question. What do you think in general of Agile? What is your opinion of it as a medium of delivering software?

[12:39] PARTICIPANT 9:

Well, I think Agile is very, very dirty word these days, dirty in the sense that everyone is using it. They call themselves Agile even though they don't practice it. I think having a correct knowledge of Agile is most important because if you don't have that, you just cannot call yourself Agile. Just by creating user stories does not mean that you are practicing Agile. You need to have proper teams of proper squads created within the teams, which clearly defines the objective of the features and how the product is going to be, clearly defined, and how the backlogs are going to be there. So, I've seen some of the augmentations, they define this sprint cycle for like two weeks or three weeks, but they don't follow it. Sometimes they release the things in four weeks because of some delays in development. Sometimes they are like, we don't need it right now. We can wait for another couple of weeks. So that's not Agile. So that's where you need to clearly define the boundaries of it. The moment you define the boundaries, it really helps you in delivering the right value. The more you plan, the faster you plan and the better you plan, the outcomes come with high quality.

[13:59] RESEARCHER:

Why do you think companies struggle with implementing fully or truly Agile?

[14:09] PARTICIPANT 9:

Sure. Well, I think the biggest problem is with the estimation techniques. When it comes to estimation, people take a lot of random guesses. They assume that this particular story will take this much amount of time and they don't use the methods or take cue from the base that harness times from how long the story took. And based on that, then whenever there's a failure in terms of estimation, then everything gets compromised.

[14:43] RESEARCHER:

But software development is difficult to estimate. It's not a straightforward type of thing to estimate.

[14:55] PARTICIPANT 9:

Of course. So what happens is when you define those story points and first time you are using, so if you are using it for the first time, you need to have some buffer in your mind when and where you are creating this story points. And you need to keep on learning. It's an isolated method of learning as well, like this implementation. So that's where the problem comes in where people think they like basic timelines which are given to the development team and they have no idea how to implement it. But then at the end of the day, they come back and say, we are far behind when it comes to the delivery. So have some buffer in mind when you are delivering it for the first time. And the more and more you have the prediction of the work that's being done or the amount of time it takes to deliver a particular story point, then your estimation becomes better and you start delivering value.

[15:03] RESEARCHER:

An important question before we dive into the core topic of this interview. How do you define quality in the context of agile software development?

[15:10] PARTICIPANT 9:

Yes! An important and difficult question. I think historically, software engineering attempted to quantify quality. But in practice, we still struggle to measure it. I'll answer this question from my experience, right? The most important aspect of software quality is that the product meets the business needs and obviously and most importantly free of defects. We achieve this by using a combination of engineering and process practices. From the process perspective, we always review the way we work and thrive to improve it. You cannot get the process right from the first go at it. It is a journey. Quality take place from the moment we write the User Story to the delivery of the code. It is ongoing review of our artifacts to assure their quality.

[15:48] RESEARCHER:

Ok, let's move to the next question. Can you choose any one of your previous experience and talk to us about it in this question? Can you describe your Scrum environment? Again, choose any one of your experience you are comfortable to talk about?

[16:08] PARTICIPANT 9:

Sure. So, I take example from [Deleted to preserve the participant anonymity] where Scrum was used in a very, very big way. So, I'll give you an example. So, e-commerce companies have a particular, just like Amazon, you have a prime date. So, we had something called the Diwali as a festival in India.

[16:28] RESEARCHER:

Yes.

[16:30] PARTICIPANT 9:

Where you have these big scenes coming from e-commerce space. So Snapdeal was also planning to launch a couple of features. These features were big features, which can actually impact the business to a great extent. So, it came up with one of the projects. I'll give you an example. So, the project was whenever a new customer is coming to buy a mobile phone on the site, they would like to give their old phone back as an exchange. Now, this old phone needs to have certain checks, if the Wi-Fi is working, if the battery is there, what's the condition of the old phone? How do you identify that?

[17:12] PARTICIPANT 9:

We need to build in an SDK within our mobile application to test this. So, there were different tracks right from the pricing point, from the listing page of the new mobile phone. So, there were 16 different tracks that were being impacted. Now, this kind of complexity when you have to plan releases of 16 different tracks, that makes it really, really difficult. So how are you going to do that? So, the scrum master plays a very important role because we have these squads who are working for different tracks and they need to be in sync with each other when it comes to the guidelines. So, one track is dependent on the delivery of another, then only they can start the work or WPIs and everybody can start working. And this can be done in parallels. So, we come up with this strategy, like how much time we have to launch this feature. So, we had to two months post-testing to launch this particular feature. And there were around 12 different tracks to 16 different tracks, which needs to be delivered within this period of time.

[18:17] PARTICIPANT 9:

So how we came up with, we discussed with all the different small squads of these tracks with the product owners. And we came up with a plan that what all tracks can work in parallel and what all tracks have their dependency. So, we mapped out all the dependencies in the beginning, and started working on the ones which can be done in parallel. And we can define specific milestones of everyone, that where we are and how we have achieved it. So that we can check the velocity and we can track it if we are running behind to be able to achieve our milestones. So stringent checks, keeping having good milestones and daily stand-ups that really helped us achieving our goal. So, we were able to deliver this three days before the scheduled time. So, we got three days of buffer with good planning and good estimation that we were able to deliver this project. That was one of the really big success of an Agile project with 12 to 16 different tracks working together to deliver one feature. Which would not have been possible if we would be working through a waterfall model or Kanban.

[19:33] RESEARCHER:

Explain to me the buffer, the concept of the buffer. How do you use it? Do you use it for testing? Do you use it for bug fixing? What do you use it for? So, you end up with three days at the end. What do you do? You party?

[19:55] PARTICIPANT 9:

So, product managers don't party just before the launch of the project. So, they wait for the success of the project. So, jokes apart, when it comes to buffer, estimating before is something that you can have like 10 percent of your time if you're implementing Agile for the first time in your organization or projects. That's when you can have at least 10 percent of buffer. Now, this can vary from 10 to 15 percent because development teams are not doing development for the first time. They might be doing development from Agile methodology for the first time. So, you keep that learning curve as a 10 to 15 percent off. You can call [inaudible] or you can call leakage for the type that can happen in that estimation. So that's something that you should have when you're implementing Agile for the first time. Or when it becomes a little mature, for example, you have already have like four or five for deliveries of four or five sprints of Agile, then you can reduce this buffer from, let's say, fifteen percent to twelve percent, or twelve percent or eight percent or five percent. So slowly and steadily, you remove this buffer more and more. With Agile it becomes a little bit mature in your predication.

[21:13] PARTICIPANT 9:

So, I'm now coming back to the second part, which if you have some buffer left or you have more leeway to predict. What do you do? So, what we do is that we have more often you need the spaces and we have those bug batches. So, it depends on the size of the project. So, like, this is a big feature like 14 different tracks. No matter how much testing you do, there could be one or two bugs that you can slip up. OK. So how we used to do it, we used to organize the bug batches within the company. So, you can go back to users, have a small round of testing. Also, you can have like testing or experiments. You plan those experiments during that time because that will help you going through if small tweaks are needed or there some low hanging fruits. This can be optimized. So, for example, performance, you can see that the more users are coming at a particular point of time, how it's going to work. So, all these months marked tweaks, you can use it if you have some spare time left, but then you have base time.

[22:18] RESEARCHER:

Ok, fantastic. That's takes me to the next question. What do you do to assure software quality in this Agile set up?

[22:30] PARTICIPANT 9:

Sure. there are different teams that work. You have a squad where you have where you have a scrum master and you have a development team. So, these are the three major stakeholders that happen except the business. So, what do you do is every track needs to define the quality parameters? So, when it comes to Agile, it's like mutual responsibility of everybody to showcase quality. Quality in terms of planning, quality in terms of delivery, quality in terms of testing and quality in terms of the overall outcome of the project that you have anticipated when you created a business requirement document. So, every team defines those quality parameters initially. That's one thing. OK. Second thing is when you

create those audits for these things, for example, then the development team is writing the code, it needs to have comments in it. That's basic hygiene for a development team. So, are they writing quality code or not, which can be understood by someone else. Is that a sufficient amount of unit testing that's being done by the development team or not? From the product code perspective, how do you prepare the user acceptance test cases, which can be used the QA team. Are most of the scenarios or cases covered and they have been explained?

[23:55] PARTICIPANT 9:

Now, another aspect of it is this has to be vetted by the third parties. So how we used to do it is we used to have these audits from different teams. So, once we had to deliver a project after opening the sprint, we have this audit committee, which is going to go and check from the development perspective that the code is of quality or not. So, it's really from the business perspective if all the requirements have been drafted in a proper format or BRDs or not. Similarly, we have these audits in place, which helps us finding the gaps and improving it from next set of deliveries. So, the third party, the teams which are not actually having any stake in your product were the ones who used to audit it. That makes it more stringent and better for the company.

[24:47] RESEARCHER:

Something came up to my mind, which I ask it as a follow up question. What you've described is a maturity of a software development process. This has nothing to do with Agile. So, it could work with any process?

[25:05] PARTICIPANT 9:

Of course, so Agile is no different. Agile is also a kind of process that you are developing. The moment we start taking Agile as something unique, we actually diverge from the course. So, what we are doing in the software development like lifecycles, we are making Agile a part of it. A lot of them, saying that Agile is something different, which is going to change the overall lifecycle, are analysts basically a kind of an injection that you give the different stages of software development lifecycle. You'll start having more iterations, you start having more reviews, you don't just finalize the requirement in one go and wait for two months or three months to get the delivery done. So, these are the different phases where you introduce the elements of Agile. You just can't say that it something totally out of the software development lifecycle. It's part of it.

[25:57] RESEARCHER:

You mention audits. Can you describe that for me in details? How do they happen? To what stage, et cetera? Because that's very interesting. I haven't heard somebody yet talking to me about doing audits during the journey of a user story or during the process.

[26:24] PARTICIPANT 9:

Sure. So, what we used to do is we used to have these audits. So how we used to do it, for example, let's take tracks. So, the product order of one track. So, every track has different teams where you have an engineering team, there's a product team, so now the scrum master has certain guidelines they're supposed to follow for every project. So, whenever a sprint gets completed for example, there's a project plan need to be created by a scrum master. Then stand-ups need to be conducted. So, all these things, which are in the guidelines of any project to be completed, you have different teams who go back and check that. So, is there sufficient documentation or not? If there's sufficient documentation, are the guidelines being followed or not? If the guidelines are being followed, to what extent have they been followed or not? Similarly, when it comes to product, the product owner has to create the product requirement document. Are all the flaws supposed to be covered - are they covered in the BRD or not? What kind of issues that came in during the time of testing of that product?

[27:39] PARTICIPANT 9:

How many changes that happen in the overall lifecycle of the BRD during this sprint. So, all these things you came up with the numbers and you map it with ideal numbers. So, for example, they get me maximum five percent of the medium level bugs, or one percent of the showstopper bugs and you define that. That could be the maximum stance as a guideline. Now, what does that mean? So, when the sprint happens, how many new bugs came in from the UAT perspective? Similarly, the product owner was supposed to bring all the documentation to work out things he missed out or the things he covered? Such kind of things. Similarly, unit test cases in the development team. So, you have 85 percent of the code that needs to be unit tested. Now, what is the coverage? Is it a plus 90? Less than 90. What exactly? So that defines the overall, the particular team that how they are working in terms of the guidance.

[28:46] RESEARCHER:

So, do you do the same thing for the code itself? Do you review code within the team?

[28:57] PARTICIPANT 9:

Yes. So, there is definitely code review that happens, right? But what I'm talking about from the audit perspective is a review that been done by the different teams, not from the code that they have written but the practice that they have followed or not, like commenting, code coverage, these kinds of things.

[29:21] RESEARCHER:

I've noticed you didn't talk about testers and their role in the process.

[29:28] PARTICIPANT 9:

Sure. So, every different organization might have a slightly different process of testing. In my current organization, we don't have a specific QA department. So, what they have done is

they created a small bucket within the engineering team because of the less amount of stand up. They cannot afford to have a separate QA team. So, what they have done is they have created some of the sources which can give 50 percent of their time to development. But they are not on the same track. So, if they are doing development, they cannot do QA for sure.

[30:05] PARTICIPANT 9:

In an organization like [Deleted to preserve the participant anonymity] audit process, we have a dedicated QA team. But these QA teams have to sit at the time of requirement gathering also. So why? Because they are the ones, when the product owner is writing the BRD, they have to sit with him to understand the business cases. And they design the test cases then and there itself. So, by the time development starts, the BRD is already there to create their own test cases. So, they don't have to wait for the delivery to come and then they have to say, OK, this test case is missing or that test case is missing. So, to avoid any kind of surprise, they had it all at the beginning of the development itself and the sprint is about to start. So, they work in tandem with the product owners.

[30:52] RESEARCHER:

So that early involvement [of the QAs] definitely has some advantages. Can you talk to me about those advantages because this is not common is other software methodologies to engage the QA early?

[31:10] PARTICIPANT 9:

So, the best benefit is saving on the timelines. So, for example, let's say in a sprint of two weeks, you have created three days of QA before the release. Now, let's say there are too many bugs that are coming up that you have not planned initially. So, if QA comes late, the chances that a lot more bugs would have come in. So, the more number of bugs that have come in, the higher the chances it will take more amount of time to fix them, which makes it difficult for the timelines to be met for the delivery. And it impacts the overall lifecycle. So, the good thing is, if you involve QA early you can predict a lot of things that needs to be tested. So, the pressure on the QA team gets reduced from getting involved at the end to being involved in the overall lifecycle is that they can focus more on the other side and functional requirements at the end rather than just focusing on the bugs.

[32:17] RESEARCHER:

So, this empowers the testers from the start. They become part of the requirements, they become part of the development as well and this empowerment makes them valuable team members.

[32:39] PARTICIPANT 9:

Yes absolutely. And not only that, QA is actually very important because UAT cannot be compared to QA because UAT happens at a high level. There's a test of say a call number.

The phone number in certain countries it might be 10 digits, in certain countries it might be 11 digits and then you need to have all those things and combinations in testing if this is valid or not. But in UAT, you are only going to test a happy case or a faulty case, you will not test all the cases in UAT. That's part of it, right. So, QA has to do all of this testing and they need to see that not only this particular feature gets delivered, it does not break the previous ones. So, the integrity is also their responsibility, non-functional is their responsibility, performance is their responsibility so there are a lot of things they need to take care of when handling a particular build. So, when you involve them at the end, they are just waiting, waiting, waiting for development to be completed and then starting to work. So, it does not give them enough time to plan out things well. This gives them enough time to plan out things well and it also improves the quality of the delivery and at the end of the day, the timelines and the overall product gets better.

[34:15] RESEARCHER: You mentioned UAT. In this example, who does UAT? The business or QA?

[34:22] PARTICIPANT 9:

So in this case, the UAT is being done by the product owner along with the business team as well. So that's how to do it because business is the owners of it at the end. They are the users, they are going to use the product but then the product owner is like a bridge between development and the users, so you have a different phase where the product owners do one round of UAT and then hands it over to the business team in a given time.

[34:57] RESEARCHER:

Isn't that a little bit of waterfall because my understanding is that they are not part of the Scrum team and UAT happens at the end? Isn't this a little bit of waterfall?

[35:14] PARTICIPANT 9:

So, like I said when it comes to Agile, you cannot have like this is the best possible version that we are using. For example, if there are big business teams involved in an e-commerce company like [Deleted to preserve the participant anonymity] where there are too many people working on the business side and there are too many product owners, that business does not have enough bandwidth to do all of that. So, product owners need to take the responsibility and hand over something which they can actually test it and you have to put yourself in the shoes of the business and do the hand over to the business. Because there, business does not have enough time to give it to you and then this needs to happen at a particular time. So, in this case, business gets involved in parallel on the side, not as part of the core Scrum team but checking it with the product owners to understand what's happening. Is it something that you need to do, ok we are doing these many changes on the business side. So, you can have a clear understanding from the product perspective that what exactly is business looking forward to.

[36:21] PARTICIPANT 9:

When it comes to the other side where the teams are small and business is very actively involved, here business is the final go ahead. They are the ones who give the final go ahead because they have enough time to do the testing and they can say, ok now this product is fine for me to use it. Especially in the case of B2B, where you have businesses not with you. So, you have customers and sys managers who become the bridge between the product team and the business team. Because you cannot do AB experiments because you cannot do back the releases like B2C. Because in B2C, it's very simple. You do experiments with millions of users or you can do AB, or you can do multi environment testing but B2B is a very different ball game altogether. B2B is something that you cannot roll back. Even though you want to be Agile, you actually fall more towards of a waterfall or Kanban than being completely Agile.

[37:21] RESEARCHER:

OK that brings me to my next question. Do you think this Scrum setup produces quality software and how?

[37:38] PARTICIPANT 9:

Well, I think like I said, it depends. It depends on two ways. One is basically how mature you are in terms of using it. So, if you see that you started using Scrum in one team, and you see there a lot of gaps that are coming, you are not able to deliver it on the timelines as planned. You are not able to estimate your stories properly, and there is frequent mismanagement happening from the release perspective or from the building perspective of the product. If you see that and you are now taking enough steps, it's going to pay for sure. So, you don't move from waterfall to Agile like from tomorrow onwards we'll start working on Agile.

[38:26] RESEARCHER:

I've seen it happen.

[38:28] PARTICIPANT 9:

Yeah so, it's has to process phased. You move from waterfall to Kanban and from Kanban to Agile and that you have to monitor. That waterfall go when new items when we can start doing it then low to medium to medium to high. The more you do it, the better you have chances of success. Otherwise you are ready for a mess and that's where it's going to fail. I've seen it failing in a lot of formulations. I will just give you an example to one of the questions that you asked, where it was failing. In one of the projects where everyone was like, we have to use Agile, it's the next best thing so let's not do waterfall, let's do Agile. So, the requirements were very clear from the beginning. There was no anticipation of changes for the next six months. Business wanted this in purpose of delivery. So, waterfall was something that was totally in place for it because there are no changes in the requirements, but everyone was like, let's use Agile. So what happened was that by using Agile, it wasted a lot of energy of the teams because you started having daily sprint meetings, you started having a lot of discussions so all that manpower of effort for discussing things, because there were no changes in the requirements that were happening. So, there were no changes in the product backlogs, so you actually end up wasting a lot of time and losing efficiency of

the team by practicing Agile rather than having waterfall where it should be. You need to be very particular about what's the exact use for it. If you are building something which needs to be changed frequently you can go for Agile but if something that needs to be static for a very, very long time, you don't need it. So, don't use it just for the sake of using it. To answer your question. It does help quality when implemented properly and the team has the right level of maturity.

[40:20] RESEARCHER:

So, my conclusion from your answer is the software development maturity of the team itself has to do a lot with quality, right?

[40:30] PARTICIPANT 9:

Yes because at the end of the day, why was Agile produced? To release faster. That was the first aim, we need to be flexible, we need to be fast, we need to be as quick in terms of taking our first product out. Let's take out MVP and keep on iterating, let's see how the user responds. Based on the data points, we are going to create our backlogs and the backlogs we keep on refining itself. So, when you don't have clarity about the long-term thing about how it's going to mature. You start giving way to work on the basic thing I need to launch and based on that, I'm going to define my requirements and build something more. In today's world of Internet companies inspired to use Agile because the users were asking for different things in a specific time.

[41:31] RESEARCHER:

I'd like to talk about something which is very important in an Agile environment is collaboration. So, it's based on highly collaborative teams and there appears to be a behavior that occurs in these teams. What's your opinion about it? How does it enhance the process or how does it constrain the process etc? For example, I worked a lot in waterfall until I left the industry. In waterfall, what I've noticed, they always look at the QA as the less important set of skills and the less important role in the process. They look at them a little bit down because they believe they are under skilled etc and there is a power struggle between the developers and the tester most of the time. So, can you talk to me about this peer-to-peer engagement in Agile using your Scrum experience?

[42:44] PARTICIPANT 9:

Of course. I think you raised a very valid point because if you remember during the waterfall days, the QA were paid less salaries than the development team.

[42:55] RESEARCHER:

Correct, yeah.

[42:57] PARTICIPANT 9:

This was for the sole reason because the thing that these guys were not doing the major work, the major work was being done by development. But if you look at the Agile setup, this is not the case. The reason being of the involvement in the engagement that QA brings on the table. Now, the companies are realizing the importance of not just the delivery person, but the quality of delivery. Because if the quality of delivery is not good, if it is not being tested properly, if the testers have not created the correct use cases and they've not tested it, then the effort it will put changing the things again is very, very high and it's going to impact the business quite a lot. And it will also impact the customer experience. So, the more and more people have come towards understanding the importance of customer experience, they cannot have a buggy product out in the market. And who is going to make sure that your product is stable, your product is bug-free, is the QA because that is the person that is going to test your product. If that person has not done enough testing, you'll keep adding the deliveries on the Agile experience. You'll keep on doing that. The more and more the problematics sprint is going on, it will have a big bucket list from the bugs. There will be a big pile of it. You won't be able to manage it and that's where the importance of peer-to-peer traction comes in the picture.

[44:29] PARTICIPANT 9:

But the QA comes and gets involved directly with the product owner and not just the development team. Because they are the ones who understands the product owner is someone who is a technically functional person. The collaboration with the product owner helps better understanding of the requirements and expectations and hopefully more efficient testing and less bugs. He or she will share the perspective that this is what we are trying to achieve. By achieving this, what other things can be tested. QA will come back with those things, be it from the functional side, be it from the non-functional side, all from the relationship perspective. This feature might impact the feature we are releasing, pre-sprint stuff. So, we need to make sure that when we are going to release this, it should not break the other one. So, the initial fixes need to be planned, functional test cases need to be clearly defined, it needs to be vetted along with the product owner, and once development starts, they get involved with the developer's side. So, the high check between different teams in the initial lifecycle needs to include QA and at the end of the day, makes Agile a better option for use rather than just waterfall. Because if you don't have that, the whole purpose of Agile will get lost. So faster delivery does not mean it's a case of Agile delivery. It needs to be fast and stable and good quality.

[45:45] RESEARCHER:

Fantastic. My next question is can you share with me a positive story about Agile? And if you can make it link to quality and achieving quality would be even great.

[46:01] PARTICIPANT 9:

Sure. So, I go back a little. We were building a trading engine for an [Deleted to preserve the participant anonymity], a normal energy trading company. Now, what happened was once we

delivered that trading engine, we found out that all the traders - because trading happens in microseconds - every user has certain kind of aspects so they use different specs and feeders and one size does not fit all. So, what we did was, we interviewed a couple of traders that how would they want it captured. They were looking for something very customized so how to do that? If you create a wonderful scenario, then we have to plan a complete release, we gave the requirements, we planned the release, but we don't have enough time for experiments in short sprints. So, we went back, and we said OK, let's try busy stock traders. We created a small Agile project with these traders and what all use cases we could come up with. So, we come up with the short sprints, we make the sure the business will not get impacted. So how we did it was that let's say they are working for eight hours, we saying we're going to expose the new user experience for one hour in a day. So that's should have no impact on the business. So, for seven hours, they going to work on the previous methodology and for one hour, we are going to use this new product. So, we came up with this short sprint of two weeks that we said let's do it. And we were able to create something different and we were able to test it and deliver it on time.

[47:51] PARTICIPANT 9:

Now looking at the process perspective, what all checks we did that we could not have done in waterfall. If we had done the waterfall, we would have needed to involve all the stakeholders, create a complete project requirement, deliver it and then start testing it. What we did was, we started the users as our QA person. We created this pool of people who are going to give testing results because this is what the user experience will be. So, we created different UIs, then started doing ABs with different users. We kept on changing this every single week post-iteration and then take that thing back and see what is working, what is not. So that tells us that creating a complete refinement which can be hold out. So that was one project that we did in a highly Agile manner which gave us really good results. So, the overall improvement in the outcome of the training sessions works around eight percent so you can imagine what business and buck that could bring. And this we were able to do in a span of one and half months. So, if we would have gone to the waterfall, this would have taken six months.

[49:12] RESEARCHER:

Ok fantastic. Now, everything or every time, not all the time goes perfect or rosy, there are sometimes things that fail and, in this context, can you share with me a negative story?

[49:31] PARTICIPANT 9:

Sure, so this happened during one of our releases. We are working for a client called [Deleted to preserve the participant anonymity], an Australian telecom company, and we were creating an e-commerce portal for them. So, we had certain timelines that we proposed them, that we would deliver in. But what happened was, there was an actual calamity in Asia-Pacific where some of the development teams were working. There was a tsunami that had happened. So, our development was impacted, and our deliveries were late. So, that's when we realized that when we are doing this planning, maybe to make sure that the development team that is working because when you have a faster release sprints, you don't have enough time so the documentation is a very important role. So, we did not have

enough documentation for other teams to pick up from there and deliver the same kind of work. So, this is one of the learnings, which is very important when you are working in Agile because documentation plays an extremely important role if you are working from different geographies and that can impact the whole delivery.

[50:50] RESEARCHER:

You mentioned a very good concept, which is Agile and a Scrum concept which is learning. The team learning itself contributes to the efficiency of the team. Can you talk to me a little bit about that?

[51:09] PARTICIPANT 9:

Yes. So, for example, let's say in one of the sprints, when they need to deliver sprints in a project. We go back to the teams and ask the questions, what according to them works well. For example, we go back to the development team and asked what worked well and what didn't work well. And, they said the requirements were not too clear and we have to go through and through to the product owner to finalize the use cases. They were not clear; they were not explained properly. If they were explained properly, we might have done it better. That could be one kind of thing that we got from the development team. Now, if you go back to some product teams, they said they had to sit down and explain all the requirements to the tech team, but they came back later with the same questions. So, they should have created some of the documents or take down the notes which would have helped them. So, you keep on getting different feedback, which helps you improve what your teams can do better in the next sprints which can help the other teams. Because at the end of the day, it's all about the handshakes that's happening from one team to another. The moment these handshakes are not firm, you will end up losing the plot because the whole project is going to suffer. You take down these learnings and put them in practice in the next sprints. That's where the self-learning mechanism comes. And that's the beauty of Agile. The more and more feedback you give, the better your Agile implementation becomes.

[52:56] RESEARCHER:

Becomes more mature, right?

[52:58] PARTICIPANT 9:

More mature, of course. So, communication plays a very important role. For example, one team might be really interested in having the textual documentation in place, one team might be really interested in having more of a visual implementation in place for example why don't you have more wireframes. why don't you just print a wireframe instead of jotting down the requirements in a text bar. So, it depends what the other team accepts because some people work in a visual format, some people work in a textual format and its very subjective. So, you need to find your own mojo, which can help you achieve the best possible results for your project.

[53:42] RESEARCHER:

Ok fantastic. That brings me to my last question. It is a little bit provocative but the purpose of it is to get your opinion and to get you talking. What do you think of this statement: Agile produces poor software?

[54:07] PARTICIPANT 9:

I would tweak it a little bit. Poor implementation of Agile produces poor products.

[54:17] RESEARCHER:

I cannot agree more with that more.

[54:21] PARTICIPANT 9:

So, I think it is more to do with the implementation, how you implement it. Agile or Scrum in my experience is not something like a magic wand where you can say tomorrow, I'm just going to implement my project in Agile and it will be a success. You need to have proper milestones, you need to have proper quality checks, you need to have proper training and you need to have proper documentation. Once you have these things in place, your product is not going to be a failure. And you need to understand one thing. The first time you going to implement Agile, it's not going to be the best implementation. But you have to keep on iterating, keep on learning so that it becomes mature and so does your product lifecycle and the quality.

[55:11] RESEARCHER:

You mentioned a process quality of course, the maturity of it. There is another element to which I would like your opinion too is the people. The people who are working on the process, they make the process either good or bad. What do you think of that?

[55:33] PARTICIPANT 9:

Well, I think at the end of the day, everyone wants to contribute. Now their understanding, their time to learn could be different. That needs to be recognized and accepted the way it is, so that it can be tweaked for the betterment of the product. So, you need to make everyone on the same page. One Scrum team or one squad cannot work in silos. It's not something that you can say, OK, I'll deliver my part, but I don't care about the other. So, that's where the feedback plays a very important role to bring people on the same page because everyone has a different expectation. For example, the development team might say I need visual, product team is giving them all the time textual because if there is no communication or feedback, nobody will get to know. So, the people have to be on the same page as the company. The company gave their share, their learning, they share their issues and they

share their success and failures together. That being the best of Agile practice. People are very important, software development does not happen without people.

[56:50] RESEARCHER:

But this requires a great level of transparency, isn't it?

[56:56] PARTICIPANT 9:

Well, if the project is transparent, it's bound to be more successful than the ones which are not. Because transparency leads to openness, transparency leads to more questions. More questions lead to more answers, and more answers leads to less bugs and better quality. So, the better the clarity that they have, it will reflect on the product.

[57:24] RESEARCHER:

Correct. So, how does transparency contribute to better quality?

[57:27] PARTICIPANT 9:

From my experience, when the environment is transparent, the quality of each one on the team is subject to scrutiny and review, nothing is hidden. Developers give feedback to each other, QAs test and provide review, etc.

I think that's how it so, it's like a chain. Just like you form a human chain, you have to form this chain of processes which you are trying to integrate when you are making Agile projects. If you don't do that, it's a recipe for failure for sure.

[57:46] RESEARCHER:

Fantastic. Thank you very PARTICIPANT 9. I really enjoyed it and it was very interesting. Before we conclude, do you have any questions for me?

[58:01] PARTICIPANT 9:

No, it was a great perspective from my side as well. It was really nice talking to you, so I got a new perspective of how you think as well. It was very good.

[58:12] RESEARCHER:

Ok thank you very much. Just procedural things, we do have quality checks as well. What we do is, we transcribe the interview and we like to validate it with the participants. I will send you the transcript just to make sure we didn’t misquote you; we didn’t claim something wrong about you or we didn’t put anything correct in there. Just skim through, if you are comfortable with it just reply to me in an email or a message if you happy with it and everything is ok. I will do that in the next couple of weeks. Thank you very much.