[00:44] RESEARCHER:

Hi PARTICIPANT 6, how are you?

[00:47] PARTICIPANT 6:

Very good. How are you Researcher?

[00:49] RESEARCHER:

I'm very well. Thank you very much. And sorry about the hiccup. The other room was giving me a problem.

[00:56] PARTICIPANT 6:

No, that's fine. I'm sorry about the delay that happened earlier.

[01:02] RESEARCHER:

No, it's okay. I understand. No problem.

[01:23] RESEARCHER:

Okay. Let's start with the interview unless you have questions for me.

[01:28] PARTICIPANT 6:

I think the only one question is what's the title of your thesis?

[01:33] RESEARCHER:

It's not a thesis. It's a paper because I finished my PhD so the title of it, we don't know what's the title of it. I have a working title is...just give me a second. I forgot what the working title is. The working title is 'What does quality assurance mean in an Agile world?'

I think we should drop the cameras because there is an echo in the voice. Hopefully, the sound will improve. Fantastic. Much better a little bit, isn't it?

[02:23] PARTICIPANT 6:

For me, it's fine. As long as you happy.

[02:26] RESEARCHER:

Yeah. Okay, let's start with the question. Can you introduce yourself and tell us a little bit about your experience?

[02:35] PARTICIPANT 6:

Yeah. Sure. So right now, I'm working as a head of delivery since the last six months here in this software services organization called [Deleted to preserve the participant anonymity] I have overall 10 years of experience and my path currently is very relevant into Agile leadership, building up teams in a way that are up to kind of deliver in an Agile fashion. In my current role, I am accountable to make sure that those deliveries do happen timely, no quality issues come out of there, the scope stays under control, costs stays under control. And the same time, I work a lot in kind of improving capability iteration by iteration for the scene. So also, part of my strategy right now is to increase Agile maturity in the teams that I'm working with right now.

[03:46] PARTICIPANT 6:

Prior to this organization, which is right now a services startup where we deploy resources to multiple clients for the tenure of projects, and then move those projects away to other projects as and when required. Prior to that I was working for the last five years with the government of Dubai within entity called [Deleted to preserve the participant anonymity] that was responsible for the [Deleted to preserve the participant anonymity]. So over there as well, I was leading the Agile transformation and set up in a way to kind of move us to digitize and onboard a lot of different government departments into one channel that we were calling [Deleted to preserve the participant anonymity] as a mobile app. I was involved in that set up from the very beginning when it with this product started from a team of two to four people and it grew and grew and grew until when I left there, it was a floor of around sixty people around five different squads, around ten members in each squad, all working for one product and scaled that product. So, I was looking into a lot of process dimensions of those things as well. Before that, a lot of my experiences is leading software development teams and being a developer myself, that's where I started my career. I think in the last seven, eight years most of that exposure was within these Scrum environments.

[03:56] RESEARCHER:

Let’s start with some definitions. How do you define software quality in the context of agile?

[04:01] PARTICIPANT 6:

Software quality is an abstract and complex concept. It can be difficult to define, but its absence can be easy to see as well. I’m sure you have heard of fitness for purpose. This means the product has to meet the business needs and obviously no defects. But, agile has a view of quality which is focused on software code, design and process. In my team, we have guidelines and standards to enforce clean code. Again! It is subjective, but we have code review to help each other’s assuring clean code. In agile we believe in responding to change, that why we give great importance to a sustainable design. The business should be able to come to us for changes all the time and we should have a design that take in changes without major changes to the software. Quality doesn’t come without a process and practices. The software development process itself is subject to continuous improvements.

[05:36] RESEARCHER:

Fantastic. There is a follow-up question, but I'm not going to ask it because it fits within the next question. What do you think of Agile in general? What is your opinion of it?

[05:51] PARTICIPANT 6:

Yes, I think it's very nice and interesting question, I would say. Because first of all we need to step back before asking, what do I think of Agile in general in terms of my opinion. And I think one important question is, what do I define Agile as. What does Agile mean to me? How do I perceive Agile versus how everyone in this industry perceives Agile? There's very common misconception because I've been working with a lot of clients and I see this a lot even with people in leadership roles, executive leadership, people who do claim to be experts in Agile. And their perception in Agile is that it's just a process to deliver the scope of work into chunks of iterations.

[06:49] PARTICIPANT 6:

You just kind of sprint things and then somehow magically you start getting speed of delivery. That's one very common perception that I have seen in clients, which to me is an incomplete perception or just part of the picture, the whole big picture. To me agility is more I would say our mindset, right? If you look into the Agile Manifesto, which has those four values of individuals and tractions being preferred over processes and tools and so and so. And those twelve guiding principles, right? You would see that it is more than just chunking the work into iterations. It's a whole mindset. It's a whole set of guiding principles. It's a whole value that an Agile team has to kind of own right and then it brings those benefits differently, like the faster time to Market and increase collaboration and of course faster deliveries.

[07:45] PARTICIPANT 6:

So Agile is a mindset that if I have to give my opinion, I believe it is something valuable. It is something that challenges the traditional ways of working. I've worked with a lot of clients to kind of transform their existing ways of working from very siloed to bringing them up into two speed kind of approach, whereby you spot the Agile catalysts in the team and then form a POC for your Agile transformation. So, you show it first to them and then they see the real value. And I do hear this a lot that Agile doesn't work. We tried Agile and it didn't work for us and the very next question I ask when I hear those thoughts are what do you define Agile and ninety percent of the times I do hear this misconception that I just told you about. So, to have the mindset is very valuable. By the same time to me defining Agile is just chunking into iterations is a wrong perception and I think anyone who thinks that needs to just fine-tune what Agile is actually about to see the real value of that.

[09:05] RESEARCHER:

You mentioned earlier that your mandate is to improve Agile maturity. What is Agile maturity? How do you become mature in Agile?

[09:20] PARTICIPANT 6:

Yeah, that's a very good question. So, look Agile maturity to me. It's not a destination. It's a journey. Agile is about continuous evolution, Agile is about continuous improvement. But then how do you define that improvement? You need to put the right KPIs or the right measures in place. You can't manage what you can't measure that's what I believe. So, there are many dimensions that I look at Agile maturity from. A lot of aspects, but I think if I have to kind of group them up in broader categories, that would be a team dimension, be the environment or the hygiene factors around the team, are they able to look at it or not? You know what kind of issues they are having, how easily they access each other. So that's on the hygiene side of it. When I said the team dynamics, I've been leading the capabilities within the teams. The third dimension will be to me the product dimension or how well the product vision, or product roadmap is laid out. How will the teams buy-in, how will everyone in the team kind of understands that every single...Can I just walk to a developer in the team and ask him why is he doing what he's doing?

[10:48] PARTICIPANT 6:

And when you look into early setups of Agile teams, you will see those whose traditional issues which whereby this thought process will be challenged. A developer will be telling you that I'm doing it because I am told to do so. And that's to me the very earliest level of maturity or immaturity. So, step by step, I kind of work on those aspects, and this is just three of the five that I defined. think after product the other couple of those would be, one will be the engineering sides of things, doing best practices, the test-driven development, the automation. So those things also come step by step, you know, you just can't bring the ideal setup in the world and just get it implemented because then you can't sustain it. So, I very much push for organic growth of things and introduce those elements step-by-step, iterations by iterations and bring more maturity. One element out of these five that I measure, that I kept for last is because like I said, that's only one part of the game. And that's the Agile ceremonies as you call it, the practices in terms of the Agile setup. Going through the standups, the grooming sessions, and the retrospectives, so around those areas.

[12:22] PARTICIPANT 6:
Those also are kind of... there is a maturity level, there's maturity metrics to that as well. You can't expect an Agile team to be perfect in how they conduct standups from the very beginning or how they do retrospective from the very beginning or how they do product demos from the very beginning. So, you have to kind of let the team adapt to it, let them learn and evolve from there. So, there are many success factors that I have defined, and I lay out as a roadmap, like okay here's the baselines, here's where I stand right now, here's where I want to go. But here are the chunks that I want, I mean this Agile transformation also takes an Agile approach to it.

[13:07] RESEARCHER:
Okay fantastic. Let's move to a little bit of details. Can you describe your current Scrum environment, or you can choose any one of your experiences and talk to us about it?

[13:21] PARTICIPANT 6:

I think for the purpose of this conversation or interview, I would choose two projects that right now we are doing. Reason being, I think it will bring a lot of perspective from these two example. One project is doing is very well and the client is very happy. The other project is kind of struggling right now, I'm working to align it and tune it to the right base. I think these two together will give you a good perspective and I hope we can get good input from you in this sense.

[14:03] RESEARCHER:

Okay fantastic, thank you.

[14:06] PARTICIPANT 6:

So, the one, the client which is a telco organization. The product we are working with them on is a mobile app. The skill set that we have setup there is a Scrum master, and the development team in terms of backend developers, frontend developers and a couple of

QAs. That's one. The other project is a client in the transportation industry and the product is a web portal, an enterprise fleet management system. There we have setup a similar to the other with a Scrum master from our side, backend resources and all that. In terms of skill sets, from our organization as a service provider, is more or less the same. But the dynamics of both these clients are totally different.

[15:16] PARTICIPANT 6:

One is more oriented towards an Agile thought process. There's a good buy-in for Agile within that context and that project is going very successful. The other client right now struggles with its own division set up, the hierarchies are very traditional. There are many silos in there. And therefore, the delivery is also struggling. There's a lot of, I must say, blame coming to us as the same time, but if you map out the whole picture, I do get down to the root causes being those silos structures and the miscommunications and everything else that come along with it. The resources that we sent out to the client, when you see the whole team's structure, I think both these projects have dedicated business owners from the client side. Many different businesspeople from the client side that I'm kind of responsible to get the requirements clarified or the idea for those requirements and then interact with the Scrum Master.

[16:30] PARTICIPANT 6:

And then the Scrum master from our side maintains the backlog. In both of these projects, in a lot of projects that we do, we don’t, and I've never seen doing Scrum by the books as you say right. But there is a dedicated product owner within the team who works with the Scrum master to track the product. I think with these varieties both of these projects have dedicated project managers from client-side, and then the Scrum master from our side. This project manager and Scrum Master share the responsibilities as that of a product owner as stated in the Scrum guidelines. So, both these parties become mutually responsible for the evolution of the product backlog in the maintenance of it.

[17:24] RESEARCHER:

So, would you say that the second example, the transport client, is struggling because of pre-existing conditions not in favor of Agile?

[17:37] PARTICIPANT 6:

There are many factors with this client, but I think the Agile mindset is very important. Like I said, the successful client does understand what Agile is about. What an MVP mindset is about, what the prioritize approach in sprints look like in a way that every end of sprint, you have a potentially releasable product. And this successful client is basically happy to let go of many features that they think should be part of the project and this mindset of prototype prioritization over there. And this project which is struggling is because the same factors that this customer does not understand Agile, does not understand more of an open collaborative concerted efforts, does not understand the definition of an MVP and therefore everything becomes very important to be done within a sprint. They try to stuff a lot in and there is a lot of friction within the stakeholders of the organization itself.

[18:52] PARTICIPANT 6:

Though every other department raises their hand and says this is important for us and therefore when you try setup prioritization, you have a very common answer, that everything is priority. So, this time everything is P1. It takes a lot of struggle to streamline things and bring this mindset. While for the other client, which is very cooperative, very collaborative in terms of letting go. If you go and read the Scrum guide, it says the art of maximizing the work not done, which is letting go of what's not value and focusing on being very lean and focus on what matters. So, this other successful client is actually from that mindset to prioritize the backlog that is being maintained. For the other struggling client, like I said, everything is priority and therefore nothing is priority.

[19:53] RESEARCHER:

Just for the sake of being clear. We will use agile or Scrum in our discussion to mean the implementation of Scrum in the above example.

[20:05] PARTICIPANT 6:

Fine.

[20:07] RESEARCHER:

You keep referring to this Agile mindset, which I understand. Can you tell me in these two examples why the telco client for example had the mindset of Agile, but the transportation client didn't or struggling with it?

[20:15] PARTICIPANT 6:

That's a very good question. I'll go into the details of what the distinguishing factors are because this is a very important question to us as well. And taking on new clients, we do keep questioning to ourselves, how does the client Agile mindset look like not just in terms of the kind of capital of the relationship potential but also what makes for us a good candidate. Because as a service industry it puts a lot of reputation in this industry at stake.

[20:48] RESEARCHER:

Correct.

[20:49] PARTICIPANT 6:

What I have seen is that there's a lot of investment from the successful client side in terms of delegating this Agile mindset. This leadership conducted a lot of training programs and this stuff on training on building a lot of resources within their staff that becomes a catalyst of this kind of transformation. While the other is I would say is taking right now baby steps. Eventually, they will go there because without this kind of transformation, I believe in this current market, there is no survival. So eventually I do believe that they will also be there, but it does take a lot of top-down investment. It can't go about one person from downstream that cannot stand up and say, I want to do things as Agile, it just doesn't work that way. It takes leadership to understand. You just have to delegate the freedom, the trust of the team. It takes leadership also to understand them the lean mindset and the MVP mindset and the art of maximizing doing everything right and the art of maximizing delivering only value.

[22:12] RESEARCHER:

Okay. Fantastic. I do have some follow-up questions. The first one you mentioned that the QA team in the telco example, for example, and also on the transportation example, they are part of the team. Can you tell me the dynamic of the QA within this Scrum set up? How does the QA work within the team and how they are engaged within the team etc.?

[22:43] PARTICIPANT 6:

Sorry, what are you saying? Who interacts with the team?

[22:47] RESEARCHER:

The QA. The QA resources. So, you mentioned that the QA resources are part of the team.

[22:58] PARTICIPANT 6:

Yeah, sure. So that's an interesting perspective to it. Normally, we deploy one QA resource with one team. In this transportation client, we have two squads, which is two different dev teams on two different tracks. But there's one QA which is shared among these two. While on the other track on the successful client, the telco, there's only one service squad and there's one QA. But interestingly the Scrum master that we have deployed there also has a background in QA. He evolved from being a QA to a business analyst to a Scrum master. And I do think, and I do believe that's pretty much helping our development there because this guy is pushing for a lot of practices in terms of QA.

[24:05] PARTICIPANT 6:

Because he is very proactive and in putting those practices. He introduced the automation, he introduced the BDD concepts within the team. So, he's the Scrum Master, he's driving the QA approach within the Agile team. While on the other hand the project manager and Scrum master that we have, evolved from our dev background mainly and then mainly into project management and then to Scrum. So, he kind of moved from a traditional project management into Scrum [inaudible] and then this project is struggling. And I think the QA mindset is also, I would say is a bit of lacking over there because this PM didn't really bother much about the QA best practices. Although he does kind of takes the QA process as ticking the boxes; this yes, yes, yes, QA is there, tick, tick, tick. But he's as good as a catalyst to QA practices as the other Scrum Master is.

[25:24] RESEARCHER:

I've noticed that the QA in this setup are engaged early in the process, which is normally in contrast to the traditional methods of software development, where the QA is at the end. My question is this early engagement, does it change the behavior? Does it bring a dynamic between the QA and the developer etc.?

[26:00] PARTICIPANT 6:

Yes. That's exactly right. I agree to what you said. It does bring a dynamic or a better perspective and to me, it's very positively correlated. Sorry for using research terms. I shouldn't say this without evidence or data, but as I see it, there's a very strong correlation between engaging QA resources early and the outcomes of the delivery. So, it does help because when the two resources involved in planning from the very beginning. They have to challenge the scope. A lot of times and when I say a lot of times, I think it's more than eighty percent of the instances, where the QA have challenged the scope in a way where the scope had to change, and they had to adjust it. So that means that you kind of detected the fault very early before that happened, and therefore lowering down your cost of quality, and therefore improving the quality of the product.

[27:08] PARTICIPANT 6:

So yes, it does. It does help in bringing that and when you do the test planning earlier when a developer is kind of working on the user story and he does see a test script ready with it. And then he's coding it, he knows what he needs to prevent. So, the test plan is right there and therefore it also kind of impacts the end and outcomes. So yeah, I think I think I would agree with your notion that it does impact and it does bring about a change in contrast to the traditional practices.

[27:44] RESEARCHER:

Okay. Fantastic. Can you take me through the journey of a requirement or a user story or a feature from its inception to the release in this setup?

[27:58] PARTICIPANT 6:

Yes, in both setups, the journey for the requirement is very common. so, there's an Inception and this inception could come from many sources that someone just came up with an idea or there is a valid business need to it. I have seen worse when the boss at the top just wants it. So regardless of the business need, regardless of the user needs, it just has to be done. So there's also a source of inception but then from those sources the project managers at the client's end, we do a kind of nominate within our agreements is that's our one single point of interaction for the project and he's responsible to consolidate all the inputs coming from all the noise around. He kind of works with a Scrum master and this idea gets added to the backlog in terms of a user story. So, there's a high level user story and there's a very initial draft or the first version of this user story that just gets added to the backlog.

[29:13] PARTICIPANT 6:

Then these two business owners, which is the project manager on the client side, which is also the business owner and the Scrum Master on our side as part of the Scrum team. They work it out and clarify those requirements, they go through it first within themselves. So, they think on the pros and cons and scenarios and everything that goes with it. And then, in one of the grooming sessions that has to happen every week, depending on the roadmap, these elements, these stories are picked up in front of the development team time. When those stories are mature enough to be discussed with the development team and when I say development team, I mean the developers and QA and designers. So, there is a question and answer session, they do get back, they take the time.

[30:05] PARTICIPANT 6:

So, there's a grooming activity that takes place and then this story gets estimated. We used to point estimations for going on both tracks, we're using some points. For the successful one, there's a stable story point scaling which is in place. For the struggling one, the story point is very variable I would say. From the estimated backlog, then we pick up items in the sprint planning. The reason why I'm hesitating to give these processes because yes, we do pick up items in Sprint planning, but many times this plan does get interrupted midway in the sprint. But it's planned in the sprint. From their expectation, the ideal expectation is that the

QA starts creating test scripts for this within the Sprint. While the dev team also kind of starts coding those things. And we have to kind of create a little lag between these two elements. So that the stories are fully ready for the developer to work on in terms of the design, the UI design, and in terms of the test scripts or test plan as we say.

[31:30] PARTICIPANT 6:

And then by the end of the sprint or before the sprint demo, the ideal expectation is that tests have been conducted within the sprint. So as soon as a developer is done with the story, he throws the story to QA. It goes back and forth in testing rounds with this story. In an ideal setup, the story which is undertaken within a sprint have to be delivered in the same sprint as shown. Most of the time, it does not happen. Most of the time the stories do spill over to the next spent. Because at the end of the day, like I said, there's a lot of developers and very few QAs. So near to the end of sprints, huge backlogs gets added to the QA's queue. Therefore, there are spillovers. But yeah, that's more or less the cycle of delivery.

[32:24] PARTICIPANT 6:

Then this completed feature is demoed to the client. They might get back with change request that have to be further planned, or they might notice some bugs that might have slipped through the QA process during the demo. Also, you know the client can come with a dimension which was never thought of before maybe. But yeah, the bugs are raised during the demo session which are then planned for the for the sprints depending on the priority. Then there's a time for release and then those branches gets merge and then goes for delivery.

[33:07] RESEARCHER:

Fantastic. Thank you. During this process, how is quality assured?

[33:17] PARTICIPANT 6:

So, I think from two or three dimensions. One is QA being involved in the sprint planning. So, their questions and answers, we refine those requirements in a way that that they are clear enough for the developer to start but also make sure that every requirement gets into the quality assurance. Our QA makes sure that the requirements are stated crystal clear enough for the dev team to understand so that it reduces the risk of ambiguity and more defects. Then with the sprint, after these requirements are satisfied, the QA is expected to write test scripts which is they will go through the stories and they will create test scenarios and they will create those test scripts one by one for each story depending on the weight of the story. Some stories will have four or five test cases while the other my may have fifteen to twenty test cases.

[34:19] PARTICIPANT 6:

If it's a story as simple as I just want to login to the system, the test case it will be the email address has to be valid. The password has to be at least these characters, it should not except this kind of password for example. This should happen when I enter the wrong password or stuff like that. And so those test scripts are put in place. Now in terms of the structural dynamics, when the developers coding the story, it's on their local machines. But before the story is handed off to QA, it has to be deployed to an environment that we call in a SIT environment. That is the first stage of deployment and that's where the QA resources are expected to test the features on and then there is defects whatever comes up in that environment. When the SIT is complete for that, and we hand over the feature to for the client to verify our tests on their side. So, testing comes from their side. There's a different environment. That is the UAT environment.

[35:37] PARTICIPANT 6:

Oftentimes there are differences in the UAT environment. There comes many issues that didn't get realized in the SIT environment. And then at the end of the day there's the final environment which is the production environment. When the stories have passed through local, passed through SIT, passed through UAT, then they go to production and over there they go through sanity tests. And also, when they go from a local machine to a SIT environment, the developers are expected to perform a sanity test on the requirements. And when the requirements go from SIT to UAT, the QA is supposed to perform supposed to perform smoke tests on those features before drilling down and it goes on scripts on the UAT environment as well. So those gates kind of make sure that this heavy filtering on any every process or every commotion out there. In an ideal world and in both these projects, this ideal setting does not happen. The moment is sprint is delivered, the next sprint, the QA resource in the early days of the sprint spends time writing automation tests for those features which were delivered.

[36:59] PARTICIPANT 6:

So, there's automated regression out there and also in an ideal world, the developer is expected to write unit tests, which in both project does not happen very well. And that's where the maturity factor comes in as we discussed in the beginning. So, the unit tests are expected to be there but they're not there as mature as it should be. I think that's pretty much the entire landscape for both of these projects.

[37:26] RESEARCHER:

Okay. Fantastic. How about this the process aspects of things? For example, Scrum has a set of rituals and ceremonies in place. For example, those ceremonies do they help assure quality?

[37:45] PARTICIPANT 6:

Yes, I do believe that there is a lot of value in those ceremonies. One is it brings a very well laid out structure to communication. It gets rid of a lot of useless meetings. The agenda or the value of each of those ceremonies or those events is predefined so there's an alignment in all stakeholders that come to those ceremonies and the rules that we know. There's going to be a stand-up, this is how the stand-up is going to be conducted, this is how long it has to be. That kind of put up a very nice framework of communication, and daily stand-ups kind of reduce a lot of gaps and miscommunication that you would experience otherwise in other traditional setups. Same goes with backlog grooming when you say there has to be this much of time dedication for this grooming and this much of a commitment from development on the user story.

[38:53] PARTICIPANT 6:

So, these sessions, you essentially saving your team a lot of time because otherwise I think traditional...I think the thing here is the maturity level. Communication is going very chaotic in a way that you know developers start literally crying that all of our time is being actually consumed into being dragged to a lot of meetings and then we are not having enough time to develop. At the end of the day, the delivery time will get a big hit because of the useless events that would otherwise happen. So, I think in terms of structure those ceremonies bring up very nice, a good communication framework. That's one. The other value in those ceremonies is transparency, because of the shape of those events, things are crystal clear things are visible to all stakeholders from the very beginning. There are no surprises. Even if there are surprises, they just pop up in a two weeks' time period, so you're not talking about waiting for the six months to know that one of the requirements were missed or one of the scenarios were missed.

[40:10] PARTICIPANT 6:

Or worse that it was perceived incorrectly, it was not up to the stakeholder expectation. So, I think transparency is also kind of one of the key factors that Agile ceremonies are doing, and I think the third very important one is also best practices that are proven. Because we have seen these practices and these frameworks successful. I mean, if you look into giants like Amazon like Facebook, or Netflix, if you study their delivery models, they do it on the highest level of Agile maturity. When I spoke about the engineering portion of Agile, Amazon is rolling out, I don't know, one production release every minute or so. Netflix does it a thousand times per day, that is the maturity level I'm talking about. I think those best practices are an example to kind of look at and follow.

[41:21] RESEARCHER:

You mentioned some very good Scrum values which is communication which I would relabel it collaboration and transparency. Do these values change the team's behavior.

[41:42] PARTICIPANT 6:

Yes, they do. And again, that's where the maturity factors come in. What I was referring to again. I have seen in the early Agile projects or Agile cycle and it's very common whenever I kind of take over a Scrum team to improve their capabilities, I do notice a lot of hesitation from the development teams to attend those ceremonies, to attend those stand-ups. It does make sense because if you get into the roots of it, you will see the standups being done in a wrong way. Those standups are perceived as reporting meetings and where the Agile discipline is not there or if you do the wrong Agile or if you're immature at Agile in the early stages, you will turn those even daily stand-ups from ten to fifteen minutes to being dragged into half an hour, forty-five minutes or even, I've seen worse, one hour of time. So the developer naturally is going to hate that kind of set up, but when I jump on and when I do introduce the right value and when I do rectify those small elements with little coaching out there then you time-box those events.

[43:08] PARTICIPANT 6:

You make sure that everyone including the developers, including the stakeholders understand the purpose of the stand up and you tell the developer that you don't have to report or it's not you not being kind of accountable when you standing there. It's more of a collaboration factor. You're only bringing things for the sake of transparency. You're bringing things for the sake of getting help if you do need. So, you know these three questions that Scrum stand-up Express, what was done, what will be done and if there are any impediments. The purpose is not accountability at that moment for that particular resource. We're not talking about micromanagement here, right? So, the moment this mindset shift takes place, the moment a developer buys in why he's getting into the Agile setup. You will notice the developer himself being a very positive advocate of Scrum and contributing and participating. I think people get very excited when they see those changes happening, when they see that the shift to the real Agile happening. You will not only see the excitement in the development team, but you will also see the massive difference in the output that they produce.

[44:35] RESEARCHER:

Fantastic. Do you think this as Scrum set up that you have in place in project one and project two produces quality software? And why do you think so?

[44:49] PARTICIPANT 6:

Yeah. So again, like I said there's a successful project in this one project and the struggling project is struggling from many Dimensions. So, if I talk about a successful project for it, I think yes, I would totally agree with this notion that you know, Agile process is bringing is producing quality outputs. And the factors that drive it, the buy in is there, the clients support is there, the client's work process setup is Agile in itself. They support it. Just the fact that the Scrum master that is there knows a lot about quality, himself having a QA background. So, he brings up those practices in QA, he makes sure those things are done and those QA practices are done better. So, automation, behavior-driven development in all of that those areas. So over there I think you are seeing a lot of better quality that I would say otherwise would not happen. And I think I would strongly kind of relate the Agile process there with the quality that everyone is happy with. We are happy. We have no complaints. Our client is happy. They have no complaints. Customer is happy, they have no complaints.

[46:07] PARTICIPANT 6:

So that's going very well right now. The project that's struggling is struggling in all areas. And that includes quality. There's a lot of question marks on our quality process, there's a lot of question marks on our quality resources. A lot of question marks on the capability of the quality resources and interestingly we tried shuffling the resources here and there. And the resources that were being questioned in the struggling project, when we moved them into the successful project, actually the successful client was happy with the resources. They think that these resources are a perfect fit. So, I wouldn't kind of in the struggling project, I wouldn't blame the capability of the people or the process, or the PM mindset but looking at the whole picture. I think the wrong implementation of Agile is also bringing about an impact on quality. So, in the demo meetings are a lot of issues that come up and then this thing is questionable, surprises come up, and questions why this was not in the plan.

[47:10] PARTICIPANT 6:

How can you miss a scenario like this? How can it be slipped during the requirement and there a lot of finger pointing, blame games, business didn't do this, or it wasn't communicated well and all of that. I'm working to improve the maturity of this client and our team while staying on budget. But as we speak right now, yes, the QA process and QA metrics over there struggle.

[47:45] RESEARCHER:

Do you think this agile setup in the positive example produce quality software and how?

[48:05] PARTICIPANT 6:

Yeah, as I said before, we produce software quality that everybody is happy with. We have a common understanding of what the quality of our product should look like and we achieve it. Scrum or agile in general enables the team to work better. They communicate better. They are more empowered to working together with specific goals, meeting the client requirements for the product and quality. You can have all the testing tools and software engineering you need; but if the team is dysfunctional then the software quality is poor.

[48:25] RESEARCHER:

I agree! But what does agile or Scrum brings better to the team to enable them to achieve quality?

[48:35] PARTICIPANT 6:

I think the most important think about Scrum is it brings people to collaborate better. I used the example of Sprint planning. During this meeting, the client, the developers and the QAs discuss and align their understanding of the requirements and expectations on quality. This significantly reduce bugs; developers know what to develop. They do not make assumptions and the QAs know what to test. Here you have it; a product with less bugs, that’s quality.

[48:50] RESEARCHER:

We're getting close to the end so I can let you go sleep. Can you share with me a positive story about Agile and its ability to deliver quality?

[49:00] PARTICIPANT 6:

Yeah, so I think the positive story will come from this telecom client. And the positive factor is that you know within twelve weeks’ time, the app goes live. The customer goes live. It's been more or less one year since we are engaged with this client. We are delivering every iteration, we release every sprint right now, and value keeps getting better. I think being live going into customers in twelve weeks’ time is a positive story in itself. And that's where the value, the power of Agile. So that's the best use of Agile because one of the values or one of the benefits is faster time-to-market. So, you are getting an edge from your competition. You're releasing first. That means that you are reaching your customers first and you feeling first if you have don't spend a lot of cost before you are failing. So, you fail it fast.

[49:08] PARTICIPANT 6:

So that's a success factor that the app is live and within a year of time there have been many updates and things will keep getting added. The team is now getting into DevOps which already levels, they look into automation and so that's where we stand. The struggling client on the other hand, whereby it has been more or less six months now with this client in transportation, and the project is still not right. To me, it could have gone live in twelve weeks as well. But the matter of fact that it couldn't go live even in six months’ time whereby even our original commitment or our original engagement with this client was also three months. So, the project eventually also got dragged. We're also kind of losing a lot on there. But that's the negative side of the story. There Agile positive story and a negative story.

[50:08] RESEARCHER:

You already answered my next question fantastic. That brings me to the last question. It is a little bit provocative, but the purpose is to get your opinion rather than provoke you. So, the question is what do you think of this statement, Agile produces poor quality software?

[50:31] PARTICIPANT 6:

So very beautifully we reach here because this last question actually connects to the first question that we did this right? First of all, what do you define Agile as? We define Agile as chunking your scope into two weeks iteration. And I don't think it will produce poor software. I can write it, stamp it, sign my name on it. But if you define Agile in the wrong way, it will produce poor software. I would think of this this way that poor execution will produce poor software. Agile or not. If you are able to successfully instill in a true Agile mindset, the true Agile values. the true Agile practices in the teams and not just in the teams, in the whole stakeholder hierarchy. If you are able to instill the agility in the top down of the organization then again, I can guarantee, and I can sign my name on it that you will see an improvement in quality. And this is coming from a lot of experience.

[51:50] RESEARCHER:

Yeah, I know you've been around doing this for a long time. Thanks PARTICIPANT 6. It was very insightful and interesting interview. Do you have any questions for me before we finish?

[52:05] PARTICIPANT 6:

I wish you good luck in this research and I would love to see the outcomes. There are many question marks for me as well. So, I think you will be answering some of those in your research.

[52:23] RESEARCHER:

I did put your name, a note against your name. I will send it to you. It will be available in August.

[52:30] PARTICIPANT 6:

In August. Okay.

52:32 RESEARCHER:

Thank you very much and go have some good sleep.

[52:37] PARTICIPANT 6:

Thank you. I think we can stay in touch, right? You can get back to any time.

[52:42] RESEARCHER:

Yes, definitely. Thank you very much. Bye.