

[00:13] RESEARCHER:

Hi Participant 15, how are you?

[00:16] PARTICIPANT 15:

I'm doing good, how are you?

[00:17] RESEARCHER:

I'm very well. Can you hear me well?

[00:22] PARTICIPANT 15:

Yes, I can. Can you?

[00:23] RESEARCHER:

Yes, I can hear you very well. If the voice gets a little bit terrible, we will drop off the camera, right? Thank you.

[00:33] PARTICIPANT 15:

Sure.

[00:34] RESEARCHER:

Do you have any questions for me before we start?

[00:39] PARTICIPANT 15:

No, I'm okay. I've gone through the list that you have sent. That was pretty helpful.

[00:46] RESEARCHER:

Okay.

[00:47] PARTICIPANT 15:

I know what you are expecting.

[00:49] RESEARCHER:

Okay.

[00:50] PARTICIPANT 15:

How do you want to do it? Do you want to go through the list of questions that you prepared or...?

[00:55] RESEARCHER:

Yes, I'll go through the questions one by one and I'll guide the interview and if you can answer them in as much detail as possible, I would appreciate. Thank you. How about if we

start with the first question. Can you please introduce yourself and talk a little bit about your experience?

[01:17] PARTICIPANT 15:

Sure. So, I'll start with my education. I have done a Bachelor's in Technology and I.T and after that, I did my post-graduation in I.T. So, after doing the education part, I moved on to joining the I.T. industry. I have been part of the I.T. industry for about twelve years. So, in those twelve years, I have played several roles, obviously, when I joined, I joined as a software developer. Then gradually moved to a senior developer and then a module leader, team lead, and eventually a project manager or a Scrum master depending on whether the project is following the traditional Waterfall model, it would be a project manager. If it were Agile, I would play the role of a Scrum master. So, I have played different roles but for the sake of this interview, I will speak only about my experiences with Agile because that is the part that you would be interested in.

[02:17] RESEARCHER:
Correct.

[02:26] PARTICIPANT 15:

In the Agile thing, as I mentioned I think while I was working as a developer, I had been part of a team working in Agile mostly. So, I was a developer in the team that was earlier in my career. Then as I moved on to a leadership role, now I also play the role of Scrum master's in projects that work in Agile.

[02:48] RESEARCHER:

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

[03:02] PARTICIPANT 15:

Very subjective and it depends on who you ask. The quality of software can be defined as the ability of the software to function as per user requirements and of course no defects. We usually focus on this aspect of quality. But there are other aspects we always aim to achieve. Like good design and clean code. Clean code is easy to read, understand and easy to change.

[03:58] RESEARCHER:

Okay, fantastic. Let's start with Agile. We already hinted that we are about Agile here. So, what do you think about Agile, in general? What is your opinion of it?

[04:12] PARTICIPANT 15:

Yeah same for me. I would say Agile rather than you know a set of tools or some manuals, it is more about the mindset. And that is what I feel about Agile. It is more about the mindset and Agile works very good in certain kind of projects, but then if someone is very much bias towards Agile and want to implement Agile in all the projects it will not work. There are projects in which Agile doesn't fit into it. So, for those projects traditional Waterfall model or some other model might be more beneficial. Whereas in some projects where Agile actually fits the picture, Agile works really very well and it makes, I mean, it produces great software in good time and more importantly presents, you had mentioned about the quality aspect it. It

also produces quality software. The reason being that you are getting continuous feedback from the client.

[04:49] PARTICIPANT 15:

So, if you actually Implement whatever is written in the theoretical part if you implement it correctly Agile goes very well. But having said that the loophole is in many cases I have seen Agile is implemented but maybe people are not very much aligned with the principles and that is where maybe it fails to deliver the kind of thing it was supposed to. But if you follow the principles, the result would be what was expected.

[04:52] RESEARCHER:

Yeah. Fantastic. I do have some follow-up questions. You mentioned that Agile works in some projects and it doesn't work in other projects. Do you have an example for me when it does work in what condition and in what condition it doesn't work?

[05:11] PARTICIPANT 15:

Sure. Say, for example, if a project has a scope that is well defined and it is very simple, I want to develop a small thing and I just want to have one page with a user ID and password. That's it. I do not want any change in it. And this is very simple that there will never be any change in the scope. In such a case, there is no reason to put it in an Agile mode, right? It will never go through repetitive iteration or something. This was a very small example, but any project which is very small and simple, but the scope is fixed in those cases Agile is not needed. Another thing is when it comes to software then it is not only about development. Once you have developed something, a major part also goes into maintenance of it. So, a maintenance project, it is very difficult, in certain cases who work in an Agile mode because when it comes to Agile, you have a set of user stories. You are in a time box sprint. You know what you have taken in a sprint. In a maintenance project, you walk in one day and you find that something is down, and you have to work on it. There is no way that you don't work on it so that totally defies the purpose of a time box sprint because you keep changing the user story. So, these are the places where Agile will not work well.

[06:45] PARTICIPANT 15:

Whereas Agile will work well where maybe the customer wants to develop software, but they do not have a very much clear picture of it. It is a big project and slowly and slowly modules will be set up around, so in an iterative model if something works well, today, I develop one module, show it to the customer. They come back with feedback. I develop, I incorporate the feedback and then go onto developing the next module or whatever changes is required in the initial module. These are the cases where Agile works really well when things can go from a small start to gain the bigger picture.

[07:27] RESEARCHER:

So, I'd like to follow up on this one when there is no clear picture Agile is more suitable. It doesn't create a great deal of uncertainty?

[07:40] PARTICIPANT 15:

No, it doesn't. Why? Because that is the reason you have a time box sprint. So, when we were entering a sprint, you know what you are developing for that particular sprint. So, if I'm entering a sprint, I know a set of user stories that I will be developing in this whatever the duration of sprint might be, be it two weeks, three weeks, or four weeks. So that depends right? A sprint has a specific timeline. So, within that sprint, I know the user stories that the team would be developing. The team is aware of the user stories that they would be working on. Now as the sprint ends as for Agile, it is mandatory to have a sprint review meeting. So, when you are having that sprint review meeting that is when the customer, the product owner, and in many cases, the end-users also come in and they can see what exactly is happening, whether it is going in the right direction. So, it just takes one sprint to let people know whether things are going in the right direction.

[08:42] PARTICIPANT 15:

If not, changes can be incorporated very early, that is where it is useful. Otherwise, once the entire project is done, say a one-year project everything is done, and then people realize that no, this was not what was supposed to be developed. That causes a lot of effort as well as cost, so it also results in lots of wasted costs but with Agile it will happen that way. In one sprint, will you know it, at the end of the sprint, during the review whether this is what was expected of the sprint.

[09:21] RESEARCHER:

I'd like to follow up on other things as well. You mentioned that Agile produces great software. It is a little bit enthusiastic to say so, isn't it?

[09:36] PARTICIPANT 15:

It is, no actually, you see Agile will produce great software only if you follow the principles. If not, in terms of quality in what I have seen with my experience in Agile, quality improves. Why so? Because it is getting repeatedly checked. The developer would do the unit testing. There would be a QA person who do the testing, then it goes to the product owner. He would either accept it or reject it. Eventually, it goes to the client in the sprint review. He also has a chance to incorporate feedback.

[10:16] PARTICIPANT 15:

So, the quality, most of the time is not an issue. What is the issue is, in doing so, sometimes other things might get delayed. So maybe there are five user stories and people end up completing only three and the two remain. So, the three that were developed would be developed may be good, but there would still be two which were not developed thus as a whole, maybe the software is not done but the part was that was done, that is good. So, the more issue with Agile rather than with the quality, I would say is with the timeline because in Agile, the estimation is not done in hours, it is mostly done using story points. That is where people tend to overestimate, some people tend to underestimate.

[11:04] PARTICIPANT 15:

So, estimation wise and then when new changes get incorporated in the same user stories sprint after sprint, that is why things might get delayed but what is developed, there are less

chances that that developed part, already developed an accepted thing, would not meet quality standards.

[11:27] RESEARCHER:

I agree. I'd like to follow up on something else as well. You mentioned that Agile may not work when people are not aligned with the principle of Agile itself. Can you elaborate a little bit on that?

[11:44] PARTICIPANT 15:

Yeah, so far example, with Agile, it is very important that the team understands what they are working on. So, it is very important they know what user stories are, they get a clear picture of the user stories. So otherwise, they'll develop something and maybe the customer wants something else, the requirement for something. That is where comes the role of the product owner and it is the role of the product owner, a product owner is not a project manager. A product owner is not a Scrum master. A product owner is a product owner. It is the role of the product owner to get the user stories properly and to put it in writing and explain it to the team in case the team has any queries on the user stories. It is up to the product owner to clarify those queries.

[12:33] PARTICIPANT 15:

Obviously, the Scrum Master would assist the team in getting any help, but eventually explaining a user story is up with the product owner. Now, the Agile will not work, I have seen examples, where maybe the product owner is reluctant to join multiple meetings. The product owner is not available because he doesn't feel that for Agile, his availability is always necessary. So, in such cases, it doesn't work. It ends up delaying because maybe he joins once in seven or eight days. He is not available and that results in delay and that results in those user stories being pushed from one sprint to the other. That is one thing. Another example is saying, ideally in Agile, the developers should be able to pick what they would like to work on. Things shouldn't be forced in Agile. If there are five user stories, the developers actually should be able to choose what they want to work on, which doesn't always happen. Scrum masters often tend to play the role of project manager and they tend to assign user stories.

[13:51] PARTICIPANT 15:

The developer might not be wanting to work on that story. He might want something else and maybe he would have preferred. That results in maybe the output would have been better if he could have worked in something which he chose. So, these are some subtle things that. Let them take ownership of their work. Let people who are required to be present every day. These are the cases done, then it will work. Otherwise its people do not understand and then if also leadership doesn't understand, why it is required for people to be present. Or for example, some of the Agile software is pretty expensive. Now if people do not understand it and they expect that to have an Agile project, I'll run it completely in an Excel spreadsheet. I do not want to spend on any software. Sometimes it might cause an issue. These are some of the examples.

[14:48] RESEARCHER:

Yes, I understand that it requires dedication and discipline. Is that the right?

[14:54] PARTICIPANT 15:

It does.

[14:58] RESEARCHER:

You mentioned something that is inherent to Agile, which is the ability to be self-organized. So, are you truly self-organized?

[15:17] PARTICIPANT 15:

To be honest, no. It doesn't always happen. I would say, the team also. It doesn't always happen that the team is entirely self-organized. We do try to make it as much self-organized as possible, but not a hundred percent. Some teams, for example, there would be a team, a mix of seniority and juniors. There will be some people who are senior. They know their work very well. There will be some people who have just joined the industry pressures in the field. So, they might need some hand-holding in what they should be working on. So rather, they might need some help, like okay, this might be the thing which I can pick on and start working. Initially, they might not know which one they want to start on because they do not have an idea of what can be in the long run. The difficulty which it can impose. So yeah, some cases there is a little bit of hand-holding, but it is as much self-organized as possible.

[16:24] RESEARCHER:

Okay, fantastic. Let's move to the next question. Can you describe your Scrum environment? Because, my understanding your experience is mainly Scrum.

[16:32] PARTICIPANT 15:

Yes, I worked only on Scrum. When I say agile I imply my Scrum experience. I'll explain the last Agile project I was part of. So basically, I played the role of a Scrum Master. Other than that, there were a team eight developers and within these eight people there was one who was also our tester. So, this person was only doing testing and eight developers. I was the Scrum master. There was a product owner and obviously there we had a point of contact from the customer end as well.

[17:16] RESEARCHER:

Hello?

[17:18] PARTICIPANT 15:

Hello.

[17:19] RESEARCHER:

Okay. So, what the team members are and how does the process work? I think we need to drop the cameras. Otherwise the voice starts getting a little bit blurry. Can we drop the cameras?

[17:34] PARTICIPANT 15:

Yes, sure.

[17:36] RESEARCHER:

All right drop mine as well. Okay, so, can you give me more detail? How does the process work?

[17:47] PARTICIPANT 15:

Yes. So, as I mentioned, so now there is a team. Interrupt me whenever you cannot hear me. Okay?

[17:55] RESEARCHER:

Yes. Yeah so far is good.

[17:59] PARTICIPANT 15:

So, the team is of eight developers. So, these people work once they get a user story, they start working on it. So their job is to finish the task whatever is assigned and within this eight people, there is also a designated tester, who though the team also does the unit testing, but then this person the designated tester, what she does is whatever user stories have been completed by the developers. She does a thorough check of all the integration parts and with the system testing everything else. So that is the team of eight people. Other than that, there is the Scrum master. That is me. There is also a product owner who works closely with our customer point of contact, who get the user stories. And he is apart from the customer and other than the point of contact from the customer end, there are also some end-users who often know get in touch with us or often come to the review meetings, the sprint review meetings etc.

[19:13] RESEARCHER:

I've noticed that the QA is engaged early in the process?

[19:24] PARTICIPANT 15:

Sorry, I didn't get your question.

[19:26] RESEARCHER:

The QA is engaged early in the process, which was not done before in other methodologies and traditional software development methodologies. Does it change the behavior of the QA and the behavior of the team?

[19:47] PARTICIPANT 15:

So it does in a sense that one thing good that often it has happened with the team, that they develop something and then much later they come to know it was faulty piece when they try to integrate it with a piece developed by someone else. Because the developer was only unit testing this piece of code, not doing it as in the system. So, having a QA early in the system

helps that you get apart from the developer, and otherwise it would be much later, it would have been an issue. That is one thing and another thing is the engagement of the QA. So initially the QA would be in another in traditional methods, the QA would be busy only towards the end of the project.

[20:47] RESEARCHER:

Correct.

[20:48] PARTICIPANT 15:

That is when the QA does all the work. And the developers would just sit idle, and vice versa. That is during the beginning of the project, the QA doesn't have any work and the developers are busy doing things. In this case, now, it is evenly distributed so QA time is utilized each and every day. So, every morning, she can come, and she can start testing on things that have already been built. So that is a change.

[21:22] RESEARCHER:

Okay, do you think that being engaged early empowers the QA?

[21:31] PARTICIPANT 15:

Yes, it definitely does. Because for one reason is that now she has more time that she can invest. Otherwise, if all things are pushed to her at the same time, then that also might lead to a quality issue in itself that things are not tested properly. Now that she has enough time, she can actually invest proper time and also in such cases, early in the cycle the opinion that QA passes there are chances that would be more readily accepted by the team rather than if this same opinion was passed at a later point of time where because of the complexity of the team, sometimes even though it is required, maybe some team member might become a bit rigid. Because of the fact that it would require a lot of changes from his or her side to fix it. Getting it early also gives people more value to the opinion of the QA and readily accepts it and just goes ahead and makes the changes. So yes, it does in particular.

[22:48] RESEARCHER:

Do you think this setup you've been describing to me as a good implementation of Scrum and why?

[22:58] PARTICIPANT 15:

Yes, this is a good implementation. Main reason I say, is that it has the necessary...so, in Agile, some people are very important. An Agile team must have the developers. Obviously, they are the most important, they are the people who are doing the real work. And other than that, an Agile team also needs a product owner, a Scrum master. So, this team has all the necessary stakeholders. That is number one. The second reason why things work in this set up is because the team religiously and diligently follows all the Scrum rituals like the sprint planning meeting or the daily stand-up call or the retrospective meeting or the sprint review meeting. So, all these things are done religiously and another reason why it works very well is because of the clarity in the user stories.

[23:58] PARTICIPANT 15:

So, having a product owner, it helps that this person gets the user stories and puts them, so we use Jira as a software for the stories. So, the product owner actually puts all the stories and writing in Jira which helps the team. So, they can actually go through the backlog before the sprint planning meeting, and ask question to the product owner, during a sprint planning meeting about it in case they have any queries regarding the user stories that we take in in that particular sprint. So yes, having all this things set up. It is definitely a good set up. I would say it is a good implementation of Agile.

[24:43] RESEARCHER:

You talked about Scrum rituals. Can you talk to me about them in detail, please?

[24:51] PARTICIPANT 15:

Sure. So, when I say about Scrum rituals, it is basically about the main meetings that we have in Scrum. So, this is something which we diligently follow. The first one to start with is the sprint planning meeting. So, this is a meeting where everyone comes, when I say everyone, so the developers, the product owner, and Scrum Master. Sometimes the customer also joins in. So, this is where the backlog of user stories is gone through, so the team goes through this backlog of user stories. The product owner, with the help of the customer, prioritizes it as to which are the most important user stories, and which needs to be developed before the other user stories. So, there's a clear priority and then in the sprint planning meeting, the team discusses as to how much they can take in for that particular duration of the sprint.

[25:55] PARTICIPANT 15:

And obviously then in the sprint planning meeting, the story points are discussed, like how much each story would take and how much time the team has based on their availability. So, the stories are finalized that needs to be developed in a particular sprint. That is one ritual. Then there is this daily stand-up meeting, which happens every day mostly. I mean, It happens every day. Mostly, at the start of the day. This daily stand-up meeting, a few things that are discussed, so three things really. So how much work has been done from the last daily stand-up meeting till now. How much work I plan to do from now till the next stand up meeting and third, is if there is any blockers that is preventing me to do my work. So, these are the three questions that each developer is asked each day during the daily stand-up meeting.

[27:01] PARTICIPANT 15:

And then another ritual that is followed is, after the end of the sprint, or the sprint review meeting. So, this is the review meeting when a demo is showed to the customer or the end-users, whoever is present. This is the output of this sprint and to get their feedback. This other three named rituals, other than this, there is one more thing that we do as well, which is called the sprint retrospective meeting. So that is a very internal thing. That is to assess ourselves, assess the team's performance. So that is a retrospective meeting is something where the team discusses as to what the team is doing well, what the team is doing not so well, what the team can continue doing in the same way, what the team should not be continuing to doing the same. And also, each team member also gets a chance to appreciate any other team member. So, this is more or less the Scrum rituals that we generally do in our setup.

[28:12] RESEARCHER:

Okay. Fantastic. How do these rituals help assuring quality in the process?

[28:23] PARTICIPANT 15:

Okay, so firstly, the sprint planning meeting right. So, during the sprint planning meeting, each and every user story. So, they user stories are already there in the black block. In our set up, what I always advise the team, is that they should go through the user stories applied to the sprint planning meeting so that they know for sure what are the questions they have for the product owner. So that helps. So now, a set of the team has gone through the user stories. They know what is there in the background. They join the meeting. So how it helps is, whatever question that he has, the product owner answers it during the sprint planning meeting itself. So, the sprint planning meeting is the start of the sprint. So, if we have any doubt or query, that is answered at the very beginning rather than bringing up questions towards the end. That gives the team of the entire duration of the sprint, to work on something, rather than if it was the question was asked later on. This collaboration with the PO helps better understanding of the user stories. Then the team knows what to develop and test. What I have experiences, it a good collaboration with a knowledgeable PO reduces defects. We receive feedback on ongoing bases and we rectify quickly.

[29:43] PARTICIPANT 15:

Then they would have only that much period. For example, if it is a whole week sprint and the team has a question, and the product owner was not present during the planning meeting, but he comes back maybe after two weeks. Then the people only have two weeks to implement whatever he explained. But having the product owner right at the beginning, that helps the team to have the entire time box of the sprint. So obviously if someone has the proper time, then it also affects the quality. Something which is done in a rush, will obviously not produce good quality things. So, this is one way. And obviously if the requirements are clarified, that definitely helps. A clarified requirement would generally be more or less a developed software, which is what people wanted, what the customer wanted. Yes, there would be changes because end of the day, once something is developed people might view that, okay, this should have been something else or we need some other changes. That is a different thing, but the basic structure would be having things clarified early, that is the help from the sprint planning meeting.

[30:57] PARTICIPANT 15:

Next comes the daily stand-up meetings. Why this is helpful in the quality part is because of the question that is asked every day, is there anything that is blocking you from doing your work. So, each team member actually has the provision to highlight or raise a red flag in case they are stuck on something. So, they do not have to sit for one day or two days or a week, and then now come up with something that okay, this is the reason why they are blocked. Now, the entire team is present in our daily stand-up meeting and the Scrum Master is obviously also present. So, knowing whatever the red flags are, it helps it taking decisions and if obviously everything works, it is a team effort. So, if someone is blocked for whatever reason, someone else will come and help that person. And the flag is removed.

[31:59] PARTICIPANT 15:

So that is another thing for the quality part. Lastly, which is very helpful is the review meeting. The review meeting ensures that whatever is developed is actually what the customer wanted. End of the sprint, we are showing whatever we achieved in that particular sprint to the customer. So, in case the customer wants some changes, he can always request but then that request also goes in as a user story in the next sprint. So, the team could incorporate that change and then the same iteration will follow, and then again, the same thing would be displayed to the customer during the meeting at the end of the other sprint. So, this are the some of the ways that doing this ritual helps in actually having a quality software.

[32:51] RESEARCHER:

Hmm. Fantastic. Thank you very much. My next question, you already answered some of my questions. So, I'll jump right on this question. Can you share with me a positive story about Agile and its ability to produce software quality?

[33:14] PARTICIPANT 15:

So, a positive story would be like, so we got this project. So obviously and due to NDA, I cannot disclose the name of the plan whatever. So, we got this project where the development was to roll out a website where certain products would be displayed. This was done in Agile. This was a new plan that we were catering to the client did not know about our capability, but the project was done in Agile. And it turned out well. As a result of this particular outcome, that the project was eventually completed in time at this whatever scope was given to us was completed. So, the scope, time, budget, everything fitted in properly. So, because this happened, this resulted in a number of projects later on, as I mentioned, this was one website that was being developed. Now this particular company planned to roll out in several other countries. So, this was not just a country-specific thing, they wanted to roll out in different countries. And there were country-specific changes so they couldn't use the same, yeah, the basic structure being the same. There will be some country specific changes as well.

[34:48] PARTICIPANT 15:

So, they rolled out in multiple countries and because we had done the first thing well, we got all the projects so that resulted in something which came in as a six-month project. Ended up being something which is still running. So, it ended up being a [inaudible] and running on because they keep rolling it all over different markets with different changes so that I would say is a positive story.

[35:19] RESEARCHER:

Thank you very much. It doesn't always go well, unfortunately, there are negative stories. Can you share with me a negative story, please? Thank you.

[35:31] PARTICIPANT 15:

Yes, so a negative story in the sense, I would mention, is that, as I mentioned in Agile, what happens is because it is done as a story point estimation, the developers often either tend to overestimate or tend to underestimate. Now in this case, the story I'm sharing, so the backlog was pretty huge. Obviously, the customer would want the entire backlog to be

completed that that is normal. But the team needs to understand how much they can take in, but they took more than they could actually deliver. So, in the first sprint itself, they realized that they have taken things, which they have not been able to complete. But this team, surprisingly enough, this team had all senior people in the team.

[36:36] PARTICIPANT 15:

So, they say, I would say, maybe they were overconfident. They thought that okay, I will still be able to complete. So again, in the next sprint, they took again more stories than they could. So, this resulted in three poor sprints, where things were not getting completed. Obviously, the customer was not happy with the results. So, this was a negative story to share.

[37:00] RESEARCHER:

Thank you very much. That's what's been interesting. Let me move to the last question. The purpose of the question, it's a little bit provocative question, but the purpose is not to upset you, but to get you to tell us what you think. What do you think of this statement, Agile produces poor quality software?

[37:27] PARTICIPANT 15:

Yeah, as I mentioned before what I think is that Agile does produce good quality software because having so many, in Agile, the thing is getting tested so many times. I mean at first the person who is developing, that person tests it, then it is sent to QA that tests it. Then the product owner has a look whether to accept the story or not. Eventually, in the sprint review meeting, the customer would have a look as to whether this is what the customer wanted, or it is something else that the customer wanted. And if any changes required, they would ask for changes, which can be incorporated in the next sprint. So yes, Agile will produce quality software if the rules are followed in a correct way.

[38:22] PARTICIPANT 15:

If it has not followed in a correct way, as I mentioned in the negative story, if people are taking more than they expect or say the team doesn't have the right skill sets. Let me put it this way, if the team doesn't have the right skill sets to do a specific thing, those are the places where Agile might result in having something that is not a good quality. But that is not the because Agile did not work, that is more because of the situation or maybe the mindset or the skill set. As I said, if you follow Agile correctly, it will need proper software.

[39:10] RESEARCHER:

Yeah. In your opinion, why sometimes some teams struggle to follow it properly?

[39:19] PARTICIPANT 15:

Yeah, that is as I said, it is more about more of the mindset thing. Like people will take more than that is one very big problem, which I have seen in multiple projects. People tend to assign more than they can achieve. [audio breakup]

What also happens are people tend to take in less user stories that results in them sitting idle. So that is obviously a waste of effort, that also happens. And another reason is that like when someone is not being available to tell them when they are in doubt about user stories. I

mean, the product owner or anyone from the customer. So, Agile works in an environment where people are very much actively involved. That is when it will work. If all the stakeholders are available and are ready to help, whenever required that is when it will work. If any of the stakeholders are not available, that is when Agile starts not functioning that well.

[40:30] RESEARCHER:

So, it's the collaborative aspect of Agile that is very important, that's what you saying?

[40:37] PARTICIPANT 15:

Exactly. Exactly. Being Agile, all the roles are important. It is not just the role of the developer. So, the reason why it sometimes doesn't work is, maybe an organization decides that okay, this project will go in Agile mode and they assign some developers to do it. Initially, they assign a Scrum Master to just start the sprint and then the Scrum Master is assigned to maybe some other project, which is starting. So, there is no Scrum master. Initially, maybe they have a product owner or someone from the customer tells them about the requirement and then that person goes to do some other task, and it is the developers are left to themselves to complete a set of tasks. So, this is where it will definitely not work because they do not have anyone to go to or anyone to get their queries clarified.

[41:30] PARTICIPANT 15:

Or even if it is clarified, maybe someone asks a question today and it gets answered at the later part of time, after a week or so. That delays the thing. So, this is not the best example where Agile would work.

[41:44] RESEARCHER:

Okay. I will go back to your earlier answers and you mentioned that you were a software developer before you become a Scrum master. I do have a question for you because this is very interesting. Do you think that Agile improves the relationship between the QA and the developer?

[42:10] PARTICIPANT 15:

In our case, yes it did because we were getting in that it was a content management project that I was working on at that point of time. So, it did help because it was Agile, we were getting feedback from this person much early. So yes, we could incorporate it. Otherwise as I said the developers often, even if the QA finds that something is not working, the developer sometimes takes it personally. Though this is personal, this is human nature to take something personally. Because anything anyone develops, they feel that that is the best thing that has been developed. So, if people like their own creation, so they develop something and if someone finds fault in it, not many people can take it happily.

[43:12] PARTICIPANT 15:

But having the QA early on helps because then the changes are easier to make than when if I had got the feedback later on. So yes. The relationship was better, I would say.

[43:28] RESEARCHER:

Yeah, I do believe that the collaborative aspect of Agile helps bring people together. The closeness between the team's member is much better and it's really eliminates the barriers and like you said yourself, it eliminates that human behavior, which is taking it personal when for example a defect is raised. That bring us to the end of the interview. Do you have any questions for me before we conclude?

[44:09] PARTICIPANT 15:

I'm good. So yeah, it was nice to talk to you. So just out of curiosity, so your research is more based on QA methodologies or Agile. I mean your focused.

[44:29] RESEARCHER:

I'm focusing on Agile but how Agile achieves quality. So, if you look at the manifesto, the manifesto is very abstract, and quality is implied in the manifesto. It is not directly mentioned in the manifesto. And if you look at the Scrum guide, it is the same thing. The quality is implied, but it's not directly mentioned. This is a gap and the knowledge of Agile.

But the teams of Agile, Agile teams do achieve software quality like yourself said that and you said it multiple times. You believe in it, and it does produce quality when it is well implemented. Yes, I agree with you. But how is not known to a lot of people. How these teams like yourself, by applying rigorously the Agile rituals etc., the continuous inspection and the loop of the feedback, the collaborative spirit in the team. All of that helps achieving quality. But there is a gap in the knowledge, which I'm trying to address is to look for... Sorry?

[46:06] PARTICIPANT 15:

No, no I said this is interesting.

[46:08] RESEARCHER:

Yes. So, I'm looking to address by this research, is to tell people this is how quality is achieved in Agile.

[46:23] PARTICIPANT 15

Okay. Okay. Yeah, that's really interesting. And yeah, so I still say quality-wise, I think the problem would be this, but yes in all the implementation mostly there would be issues in a timely delivery, so people promise seven user stories but end up delivering probably maybe five. So, timeliness is more of a concern, at least in the projects that I have worked on. I have seen in most of the projects, timeliness is more of a concern rather than quality. And since you are looking at the quality, maybe you can also look into another thing on how quality is implemented. Just out of my experience, although Agile and even Scrum says the team should be co-located but in current scenario, it never happens.

[47:12] RESEARCHER:

It never happens, yeah.

[47:14] PARTICIPANT 15:

In the current environment, people would definitely be in different cities or even different countries, so maybe that that is one idea which you can also add to your research as and when it is about a distributed Google team.

[47:29] RESEARCHER

Yes. There are people who looked at that. But they didn't look how quality is achieved when the team is distributed.

[47:44] PARTICIPANT 15:

Because that is what one thing.

[47:48] RESEARCHER:

Are you distributed? Are you a distributed team?

[47:51] PARTICIPANT 15:

Yeah. We are.

[47:53] RESEARCHER:

Okay. So how do you manage to do the third stand-ups and all of that? All of that is done.

[47:59] PARTICIPANT 15:

All of that is done the way I'm talking to you. So, the daily stand-up is done over normal conference call. We don't have video enabled or sometimes we end up with video as well and for the other meetings, we have video-enabled calls with everyone joining in. It works fine for us. So, with the help of tools and technology, even if someone is located in a different place, they need not feel that they are located in a different place. There are online, multiple software, like I give you the example of a sprint retrospective meeting, if I would have the team in one place, I would have them in a room and tell them, okay, what is going well, what is not going well. Because it is distributed, we put it in our shared drive where people put in their comment and if everyone comments. So, yeah with the help of technology, we somehow manage it.

[49:13] RESEARCHER:

Okay. Fantastic. Just last thing before we leave, I will send you the interview transcript. If you can have a look at it and make sure it's valid, we didn't misquote or anything like that. So, this is a quality assurance thing we do. We validate the transcript with the participant just to make sure everything is okay. If you can do it, just skim through, and make sure that whatever you said it's in the transcript and yeah, let me know what you think.

[49:54] PARTICIPANT 15:

Sure, not an issue. So, when do you want this done?

[49:57] RESEARCHER:

Once it's ready, I'll send it to you, maybe in the next couple weeks.

[50:03] PARTICIPANT 15:

Okay, no worries.

[50:04] RESEARCHER:

Thank you very much. I wish you a good night because it's already night over there.

[50:09] PARTICIPANT 15:

Yes, it is. You have a great rest of the day.

[50:13] RESEARCHER:

Okay, bye.

[50:15] PARTICIPANT 15:

Bye, nice talking to you.