[01:53] RESEARCHER:

Hi, PARTICIPANT 21. How are you?

[01:56] PARTICIPANT 21:

Good, how are you?

[01:57] RESEARCHER:

Very well. Thank you very much. There is a bit of an echo. I think we should drop the videos, so hopefully the audio will improve.

[02:16] PARTICIPANT 21:

Okay. Sure.

[02:25] PARTICIPANT 21:

Hello?

[02:26] RESEARCHER:

Yes. Fantastic. There is still an echo. But I think we can live with it.

[02:32] PARTICIPANT 21:

Yes.

[02:34] RESEARCHER:

Ok. I'm just going to start by introducing myself, telling you what I do and why I'm doing these interviews and we will kick off the interview. How does that sound?

[02:46] PARTICIPANT 21:

Sure.

[02:48] RESEARCHER:

My name is [Deleted to preserve the participant anonymity]. I'm a postdoc researcher. I do my research in software quality. I try to understand how software teams produce quality using various methods [Deleted to preserve the participant anonymity] and now I have an interest on Agile. I'd like to understand how Agile produces software quality. Do you have any questions for me before we start?

[03:31] PARTICIPANT 21:

Not at the moment.

[03:34] RESEARCHER:

Alright. How about if we start by introducing yourselves and tell us what do you do? And what was your experience?

[03:47] PARTICIPANT 21:

Sure. So, I'm a QA engineer. I've been working in I.T. for the past nine plus years. I started as a trainee back in [Deleted to preserve the participant anonymity] And then I started learning on the job and change companies eventually and started learning more and then I got into quality and Agile. And when I started working in Agile projects, I thought that it was way more fun because it was more relaxed. There was a big focus on like lean documentation and things like that compared to what it was before in the Waterfall world where, we were doing a lot of bureaucracy. There was a lot of documentation going back and forth. Once the documentation was written, but hardly ever people looked at it. So, it was a nice change, to be in a more Agile, lean environment. And then just continued growing my career. And my last role is at was at [inaudible] as a quality coach where my role was to help develop the testers and how to have a more modern and Agile approach to testing. That's a bit of a summary about me.

[05:15] RESEARCHER:

I'd like to touch on a few things you've mentioned. You mentioned that Waterfall had some bureaucratic processes and rules in place. How do you think by eliminating bureaucracy, the software development process is better?

[05:36] PARTICIPANT 21:

So, I'm not saying it's bad or it's just different. Like in the Agile world, you usually have, you know, small teams and the deliveries and sort of the stuff they deliver. And I'm like, you have a cross-functional team where you can make fast decisions, you can collaborate more, whereas like in the past, it was much more of a handover from stage to stage, provide that handover so people knew what they were looking at and so forth. Like what they needed to do, like you have to have all that documentation so that the information could solve, but in the Agile world, like the information they throw in, like constantly all the time, the team is working together. So, all that handover, I work in bureaucracy, like, hey, this is the design, this is what you need to build, like in big chunks. Yeah, I just think the Agile way where you are delivering, almost every day, if not more than one times a day, and the timeframes, and all that.

[06:47] RESEARCHER:

So, your experience is mainly Scrum?

[06:58] PARTICIPANT 21

Yes, when I say Agile it is mainly Scrum. This is how I experienced Agile.

[06:50] RESEARCHER:

Great! So, collaboration replaced bureaucracy in Agile?

[06:55] PARTICIPANT 21

In a way, yes.

[06:59] RESEARCHER:

Does collaboration usually work better?

[07:05] PARTICIPANT 21:

Like, personally, I think, yes. In my first project, the only project I worked in was Waterfall. It was like I was sitting at my desk every day right, writing test cases, or some other day I would just execute test cases or someone else did, then it was just like boring work. But then in the Agile cross-functional team setting, probably two days are not the same. And you're working on what the team needs at the moment. So you're testing whatever they're building at the moment or you're helping to plan the user story to make sure that once a developer builds something, there are no assumptions, and there's no misunderstanding between what the devs understood from my user story or what the tester understands from the user story. The whole development team understands the requirements better. And what ends up being delivered at the end.

[08:08] RESEARCHER:

Ok, so the dynamic of the team you find it rewarding. You mentioned that the team understands the requirements better. How this understanding helps the team to achieve better quality?

[08:16] PARTICIPANT 21:

Yeah, I think so, because as I said before, we make less assumptions. The developers produce features that are better aligned with the business needs and the testers know what to test. This knowledge and close communication help reduce bugs. Collaboration makes a difference in software development, you're not just interacting with testers every day, which is not a bad thing, but you're interacting with other disciplines and then you get a more holistic understanding of how software work, what the business want and how software helps the users. And you learn from other professionals like in a tester and develop relationship, there's always an exchange of skills and knowledge because when you work together with the engineer you might bring attention to issues that maybe a developer didn't think about. And if you are working with a developer, maybe he has a troubleshooting tool or something to that aids in testing, which if you didn't collaborate with the dev, you probably wouldn't know there was a different way of doing things or a faster way of doing things.

[09:12] RESEARCHER:

So, do you feel as a QA or as a tester, do you feel more empowered in Agile than in Waterfall?

[09:23] PARTICIPANT 21:

Absolutely, cause you are there helping build the products and you're usually you're there helping define how the product is built. In teams that I work with, like we did all the information for the features, we had workshops, we got together. We looked at a problem that we knew our customers had, and from there and just from the problems, we would go away and decide how we build things. So, what is our vision for what the product could look like? Yes, you're right there when the work is being defined and you're working on it. But then after you see, you know, like, hey, this is an idea I had and then you see it built into the product and you see it live. It's a lot more rewarding. And you know that you are impacting the product and you are impacting that type of work that you do and how you do it as well.

[10:27] RESEARCHER:

This is empowerment in Agile, does it help you produce better quality? Does it help?

[10:37] PARTICIPANT 21:

Yeah. I would say so. I’m not sure whether this is agile or the organization I work for. But we are encouraged to speak our mind without fear. This helps a lot. For example, when I see work with lower quality than I expect or the whole team expect, then I’m not afraid to say it. If you were with a team, like if you work in a team that's engaged and people like care about what they're doing, then there's that psychological safety where sometimes you can make mistakes or other people make mistakes and know that that's okay. That's definitely a better feeling. Not only mentally, but I’ve seen people attitudes changes. They care about the quality of the product and this helps for better quality.

[11:03] RESEARCHER:

Let’s step back a bit and define what we mean by quality? How do you define quality in the context of agile software development?

[11:14] PARTICIPANT 21:

As a QA, I would say free of bugs first. Then, meets the business needs. But this is not all, right! Quality is also structural or internal. Developers aim for good code and design. It’s fundamental to agility because we like to deliver software more frequent. Saying that, if you don’t have a good process you not going to deliver at all.

[11:33] RESEARCHER:

You already indicated your opinion of Agile, but let me ask you again. What do you think, Agile?

[11:40] PARTICIPANT 21:

Agile is more like for me a kind of a mindset. So, like how you approach problems, complex problems and how you go about solving it. Agile, I don't know if we're going to talk about a specific thing of Agile like Scrum, for example. A lot of people, they confuse Scrum and Agile, they think it's the same thing. But really Scrum is just like a framework with some activities that you may want to follow if you want to get results. But what I see is that a lot of people think that Agile is just the processes and the ceremonies and things like that. And it's not. It's just like a mindset and knowing that something that helped you solve the problem in the past, may not help you solve something right now. And sometimes, yes, your past experience does dictate how you work or how you approach goals. But at the same time, if you are truly Agile like you're always open to learning new things. You're open to try new things differently. And you're always trying to get something that's a really important project that you want to be continuously delivering new features to your customers and doing so in the most efficient way. And I think just the turnaround of how software is built more quickly in an Agile way. I think that's my favorite thing about Agile. Yeah. So that's pretty much what Agile is for me.

[13:02] RESEARCHER:

Fantastic. Let's move to the next part of the interview, which is talking about a specific Agile set up. Can you describe to me your Agile environment in your last job, for example, or in any one of your previous jobs?

[13:24] PARTICIPANT 21:

Sure. You mean just the Agile set up?

[13:31] RESEARCHER:

For example, did you choose Scrum, did you choose Kanban? Because those are the framework that's inspired from Agile. And people implement them as a set up to run the agility or their Agile environment.

[13:50] PARTICIPANT 21:

Yeah. So, at my last job, we had Agile coaches and I was a quality coach and it helped that we worked together to find the ways of working. And we adopted some aspects from Kanban and some teams followed a bit more by the book and some other people they just did like stand ups every day. And some other teams, because the nature of their work was more like fluid, they did Kanban so that there wasn't like a straight, hey, you have to use this framework. It was more up to the teams to see what worked for them or if the team had skills in either Scrum or Agile. But if they even wanted to do XP programming, they could do it if they wanted. In my last job, what happened, like in general, you could see that there wasn't a lot of people that had that Agile mindset, even though they had been working in Agile environments or they were part of a Scrum team. You could feel that like [inaudible] or the way they worked or the way they thought about quality or the way they thought about monetary [inaudible] that wasn't like super Agile. So, the first thing was to help the team become more Agile, was to introduce like Scrum and kind of train it by the book with some of the teams who were not as confident in their Agile capacity.

[15:39] PARTICIPANT 21:

And I personally worked mostly in Scrum teams. We had two week iterations, we used to do sprint planning and daily stand ups and retros, and backlog planning the whole thing. It's always very different from company to company. Some people do it more by the book, with story points and planning poker and three rounds if you don't reach an agreement and some other teams, they don't even do anything as long they try and understand the scope of each task. And looking if it's too big, maybe we need to break it down into smaller chunks or, you know, like, how do we state the story. Do we do it vertically or do we do it by layer. It old varies from team to team.

[16:42] RESEARCHER:

You mentioned something a few times which is the Agile mindset. What is an Agile mindset? For example, if I will have to work in an Agile team, how would you coach me to have the Agile mindset, I guess?

[17:07] PARTICIPANT 21:

But if you join my team, we already have some set up practices. I would introduce you to that and I'll probably share some articles for you to get acquainted with Agile. Because I mean, Agile right now is very much common currency.

[17:40] RESEARCHER:

Correct.

[17:41] PARTICIPANT 21:

So, nowadays, if you're coming in the team and don't know anything about Agile, it's probably a red flag already, but if I had to introduce them, I'll probably introduce them to the basics like Scrum, which is not the Agile mindset, but it's a way of knowing how to work in that environment, I guess. And work towards building your Agile, your agility. I guess that also depends on what's your interest. If this person is interested about the process, I can probably, you know, we can go out for a coffee. I can explain how things work and why we did that way in that way. The next time that he goes into another team or project, those things that learned and enjoys working in that way, that person can see it solves problems, they can go and apply it at the next company.

[18:48] PARTICIPANT 21:

Yeah, it's not straightforward to explain, hey, this is the Agile mindset, and this is how you do it. It's about practice and then working in those environments and then coming to, I feel like I'm coming to the conclusion that Agile is just a way of working where you combine a lot of different principles, like design thinking, and lean and a better way of trying to make the way that we work more efficient.

[19:28] RESEARCHER:

Ok, great. Thank you. What do you do to assure software quality in an Agile setup? You can use the example of your last job, for example, if you wish.

[19:44] PARTICIPANT 21:

Sure. So in terms to assure quality, the way that I like to approach quality in the Agile environment is that my priority is to help the team understand what they mean by quality and what exactly is quality is for them, because it is a very subjective concept. You have a concept and you have a different understanding of quality, or you consider something to be of a quality aspect. And maybe I don't. So, the first thing is to find a common language and to find, hey, so when we see we want this to be of quality or we say this is of high quality, what exactly do we mean? Does it mean we wrote an automation tests for it or does it mean that we just going to check it. Depending on the type of work and depending on the reason the feature was introduced, it would also depend on my approach, because if you're just changing a string or text, I'm not going to run a full regression on that. But if you are doing refactoring of one of our top features where users are logging in every step to use it, then that's probably going to be more thoroughly tested.

[21:19] PARTICIPANT 21:

And then what I've been doing lately, I try work with the PMs very closely and with the devs, and everyone in the team, but with the PMs specifically to try and break down big stories into smaller ones that can be more testable, like easier software to deliver and that's how we deliver value quicker to our customers. I did the acceptance criteria to the stories to make sure that we all, as team, we understand what needs to be done for that story. And how do we achieve completeness. And which has the Done criteria, I guess. With the developers, I worked with them after they've done their coding work. We'll sit down together, and we'll do some further testing and check some of the acceptance criteria. We would probably agree on what level of testing needs to happen for story. And then after they're developed, I'll talk to them and say, OK, so what have you tested about that. And they say, OK, I tested this and tested that. So, if you tested that, we don't need to do that right now. So, let's go and focus on things that you haven't tested for some other risks that I came up with or based on my domain knowledge of the product or my own experience.

[22:48] PARTICIPANT 21:

And in that way, when we have those in those peering sessions, I get to see and read more about like their working environment. I probably get a bit closer to the code and they will learn a bit of my testers, intuition, magic. And it starts rubbing off on them. And then separately we probably organize sessions. We keep saying, let's do our regression testing. But in fact, people don't understand the regression testing. So probably do a session and explain, hey, this is regression, and this is what we do. And when you ask me for a full regression, it's actually not a full regression. What we do is we just check the most risky areas. And I guess another layer of quality in Agile, and there's not just one way to test, and there's not just one way to do things for all of the stories. Depending on the context, depending on the type of work it is, we'll have a different strategy and we have different levels of testing. So, you'll have your unit testing, you have your integration test and you have all your different tests.

[24:06] PARTICIPANT 21:

And then you have accessibility and node and performance. And depending on the type of work, depending on the type of feature or work, it's the type of testing that you do. Some of the things that we want to accelerate the time for feedback. So once the developer checks the code into a repository, how long does he have to wait to get feedback if the initial condition is correct. So that's kind of like, you know, implementing different types of automation. So, it's of the API level, UI level and organizing them in suites. So, doing the smoke testing first. And then after that, we can move the build into an environment where we can run more tests or are not, depending on what the type of work is like. Does that make sense?

[25:13] RESEARCHER:

Yeah, it does, it does. Thank you very much. I will ask a challenging question. As a QA engineer, testing is not everything in QA. It's verification. So, it's quality control. Do Agile teams practice other software engineering quality practices like code reviews, following best practices, because Agile calls for technical excellence? Do you see it in the way Agile is implemented?

[25:53] PARTICIPANT 21

If I see other types of practices?

[25:58] RESEARCHER:

Yes. Like code review. following best practices.

[26:03] PARTICIPANT 21:

Yes. So, I mean, code review is pretty much mandatory, it needs to happen by devs, it's kind of non-negotiable. And then, I guess things like monitoring the logging and alerting, testing in production, doing feature [inaudible], canary releases, dogfooding, all those are things that help accelerate the speed of delivery as well increase the confidence and increase the risk appetite as well for teams, because if they know that if something goes wrong, then we're going to get an alert and they can turn off the feature flag and it's back to business as usual. Then they'll probably take more risks and then we'll monitor our production use closely. And then instead of doing exhaustive testing before production, we put the software in the hands of our customers and then we learn from their use. If they are doing a three week regression cycle, to do it for regression, we just put it in production, let it burn, see if we get any errors. If we do, we'll roll back or turn off the feature flag, and if things go great, we say they rip.

[27:44] RESEARCHER:

Ok, great. Next question. Do you think that's Agile teams produce software quality and how? Or in your case Scrum teams.

[27:54] PARTICIPANT 21:

Yeah, they do but it goes back to that definition of quality, what does it mean, quality? What does it mean quality for them? They need to know what good quality is for them. And then that way, you don't know if they've achieved that, because if there's no definition and we have different opinions of what quality is, then we're always going to be like in disagreement.

[28:27] RESEARCHER:

Once it's defined, for example, let's assume this definition, which fits the purpose, which is the most easiest definition of quality, it fits the purpose. Which we discussed earlier. The product fits the purpose. In this scenario, I would assume quality is a less buggy product that I would assume. I would demand a less buggy product. Do we achieve that in Agile teams?

[29:01] PARTICIPANT 21:

Do we what, sorry?

[29:03] RESEARCHER:

Do we achieve that in Agile teams?

[29:06] PARTICIPANT 21:

I mean, I can speak of my experience.

[29:13] RESEARCHER:

Yeah. From your experience.

[29:16] PARTICIPANT 21:

Yeah. So, working in Agile teams, I can say yes as far as I know. So beforehand, we'll draw up our plan and we will understand what's this purpose for a particular thing. And probably we have a threshold, or we have a metric that we want to achieve. So, if we say less buggy, so what's the bug rate now, what are the number of bugs that are open. And then third, we will compare if there are less bugs or the bug rate is smaller and then OK, we say yes, we may take this to the best of our knowledge. But what I think one thing that's important about here, is that it's not just one person's opinion of, hey, yes, we deliver quality. You have the QA stamp, we're in production. It's not like that. It's a team conscious decision. Like, there are some calls, like meetings or ceremonies you can do, like Go, No Go or something like that or a prerelease whatever. And then say gather a few stakeholders in the team, team members and say, OK, we need to release this here. We built everything we needed to build, but there is a bug that could be dangerous. And that's my job that if I found the bug, just to relay that information to people that make the decisions. And then the team will weigh in the cons and the pros, and we'll decide if we reached that goal or we are ready to release, or we achieved the quality that we wanted.

[31:25] RESEARCHER:

Ok, great. Thank you. Next question. Can you share with us a good story, a positive story about Agile and its ability to produce software?

[31:43] PARTICIPANT 21:

A story?

[31:48] RESEARCHER:

Yes, a good experience, for example.

[31:51] PARTICIPANT 21:

A good experience. I can probably tell you about a recent one. So, I worked for a sub product. We run subscriptions and as soon as COVID-19 was declared a pandemic, lots of business started closing and canceling their subscriptions. So, when we started seeing that it got my team's attention by the CEO and the COO and they said, look, guys, we need a way to pass subscriptions. We want to stop charging customers for an indeterminate period of time. So that's when they instead of canceling altogether and we lose them, we keep them just by their subscription polls, we don't charge them anything. So, we were able to build the feature, define the feature, build the feature, build monitoring, alerting, test it thoroughly in three days. So, we released a whole new feature with a team of three people in just three days. Which in my team, my company, at the moment, was unheard of because we usually take more time to build things because it's a complex product.

[33:26] PARTICIPANT 21:

There's risk involved with us doing things wrong. So, I guess that was a nice way that I think we were Agile in the moment because there was an immediate need of the business and we were able to deliver that with high quality. And it was good for the company and it was good for customers as well. So, they can take a break from the paying us.

[34:00] RESEARCHER:

Ok, fantastic. How about a negative experience?

[34:04] PARTICIPANT 21:

About Agile?

[34:05] RESEARCHER:

Yeah.

[34:07] PARTICIPANT 21

Negative experience. Interesting. Negative experiences in Agile. I guess at the beginning, when I started working in Agile, I was working in a team and the team got expanded and I was supposed to help the team. But I wasn't given directions, like, hey, you need to go and talk to this person or you need to reach out to these people, or you need to go to these meetings. I think I was very much like in the air. And I remember that was very, very frustrating because I was there sitting and waiting. You know what to be told what I must do. And then I had a bit of a tantrum on work. That was like six years ago, I was young. I had a bit of a tantrum. And I was like, I need something to do, I was bored out of my head. And at that moment, my manager was like, why don't you go out and look for things to do, like define the work, you what things you want to do or how do you want to improve the process or maybe update some test cases, or you go out and look for things to do. So, I guess I was kind of a negative experience but then because instead of just being a bit more proactive and go out and look for a job, I could probably still be working with that company back then, but that kind of situation frustrated me. And I then moved to another company. It's not super negative. Like, I kind of block those out. It's just part of life and work and I don't have to agree all the time. We don't have to know what to do all the time. Like, we need to allow ourselves to make mistakes.

[36:22] RESEARCHER:

Last question. What do you think of this statement? Agile produces poor quality software.

[36:32] PARTICIPANT 21:

Poor quality software?

[36:36] RESEARCHER:

Correct.

[36:40] PARTICIPANT 21:

What I think about that statement...I think that's a very unfortunate statement. What is based on like this person that made this statement had a bad experience with quality and Agile and their team or what does that mean?

[37:02] RESEARCHER:

Well, what do you think would make someone say that?

[37:07] PARTICIPANT 21:

I guess it could happen. Like, one thing that I could say is. let's say I'm a developer. And then I am in the Scrum team and then I work, I'm told I need to work in a feature, and I go away. We are Agile, we are not micromanaging me. I don't need to tell you what I'm doing every day, or maybe I do, but let's say, I don't. And then I go away, and I feel something and it's completely not what was right or ends up being a huge problem. And no one knew it until very late. I guess that could contribute to poor quality because this person worked in a silo and didn't communicate, didn't get feedback. And then took too long to build something that wasn't asked of them, and was completely different to what was actually required. I guess people also mention, a lack of documentation, of the lean documentation approach. It also can lead to misunderstandings.

[38:31] PARTICIPANT 21:

Yeah, I guess I have written the anecdote about a manager wanted a developer to investigate how to integrate a different service with our products. This person went away for a month and built the whole damn thing, the whole integration, without consulting with anyone else. And then when they reached code review, they said it was just a mess and the developer didn't understand what this person was trying to do, didn't understand the code he wrote in the first place and it was just a misunderstanding between the PM and the dev. The PM just needed a research investigation. And then the outcome was a detailed plan on how we could achieve it. But then this person went away and built a whole integration which wasn't needed and didn't end up being built in the end.

[39:47] RESEARCHER:

Thank you for sharing that. I do like to follow up on something very interesting about you and your experience as being a QA coach. Can you tell me what do you tell the developer and the tester or the QA in the team, how do you make them work together? And what is the best way to make them work together?

[40:21] PARTICIPANT 21:

So, let's say for one of the teams, there was a junior QA. She was mostly coming from a Waterfall, big corporate background and we were more of a high growth startup. And we were somewhat Agile. And this person was very much like, this is my plan, we need to run that. And when people say, hey, this is too much, we don't need to do that. We're always like, are you saying that she thinks this needs to be done? And then I work with her and I kind of stuck her through, like, I trying to understand where she was coming from. What was her goal. I also had a conversation with a developer in the moment, the person who gave me the feedback about her and I said, you know, like probably you need to be a bit more patient with this tester. And told the tester to be more open to other people's opinions because it's not your responsibility to have everything tested to the T and everything has to be perfect. Your job is to work with the developers and then ask them what type of testing they've done or work together and do some testing together and work collaboratively. I think that's very important. I think helped me is to be a part of that, be a part of the team, be engaged with other teams to build relationships with the people that you're working with.

[42:01] PARTICIPANT 21:

And then everything is going to be being much easier. So, it depends, going back to how do I approach quality coaching, it's the same thing. Like I go into a team and if they ask me for my help, I want to understand why they ask me for my help and why they think they need my help. And sometimes the symptoms of a team may not be the actual problem. So, really talking about what they need from me or what they want from quality or what the quality needs are. It’s the first thing that I go in and discover with the team. And then they will come back and develop a plan and a strategy. Let's say this team wants to start doing [inaudible] testing. So, OK, we set a plan, let's start doing testing from the next sprint, for two sprints and see how it works. See if we are able to find more issues in the beginning or find more issues in the later stages. So, defining the plan and making sure we know how we can achieve that. And that's measurable. And then go from there.

[43:23] RESEARCHER:

From your experience, what you have noticed in practice, does the relationship between the test and the developer improves in Agile?

[43:35] PARTICIPANT 21:

I guess now in Agile, they actually have a relationship. Like, they know each other and probably the sit very closely. Back in the day when I was working for this huge testing factory, where I was just doing test execution, I didn't even know who the developers were. I literally I didn't know who they were. I didn't know if they worked in the same company as me. I didn't know who worked for the customers. I didn't know. So, yeah, I think Agile brings people closer together. And they kind of get together. A small team comes and go and they're independent and it's just a more engaging and an empowering way of working.

[44:39] RESEARCHER:

You mentioned something very interesting, which is the closeness. Does the closeness help, too?

[44:48] PARTICIPANT 21

Sorry?

[44:51] RESEARCHER:

Does the closeness of people being close to each other, does it help?

[44:56] PARTICIPANT 21:

Yes, it does. If you can and everyone is co-located, it does help. Because you just turn around and you talk to someone or you overhear a conversation that you could probably influence, or you could benefit from. You can pick those up much better. But I guess it depends on the team's maturity. If they're a mature, remote working company, they find ways to make it work as well. Like the ways we are working, the Agile way, it's not working remotely then they'll probably find a way to make it work. I guess that's a plan of being Agile, right? Adapting and adapting to change.

[45:59] RESEARCHER:

Correct. It was a very interesting interview. Thank you very much for your time. I don't have more questions. Do you have any questions for me?

[46:09] PARTICIPANT 21:

No, no questions. Please feel free to send me a message later if you have any other questions, I'm happy to answer them.

[46:18] RESEARCHER:

OK. Thank you.

[46:18] PARTICIPANT 21:

It was really interesting. Thank you very much.

[46:20] RESEARCHER:

And just a final procedural thing. We do validate the transcript of the interview. If I can e-mail it to you, have a look at it and tell me if everything is OK and it's an opportunity for you if you want to add something, you can add it as well.

[46:39] PARTICIPANT 21:

Absolutely. No problem.

[46:42] RESEARCHER:

Thank you very much. Have a good night. Wow.

[46:46] PARTICIPANT 21:

Good luck with your research.

[46:48] RESEARCHER:

Thank you very much.