[00:00] RESEARCHER:

Hi Participant 16, how are you?

[00:02] PARTICIPANT 16:

I'm doing good. Let me quickly connect my headphones.

[00:07] RESEARCHER:

Ok sure.

[00:23] PARTICIPANT 16:

Alright.

[00:26] RESEARCHER:

Well thanks for accepting to do the interview and I'm really sorry about earlier. I got the time mixed up. I thought GMT.

[00:36] PARTICIPANT 16:

No, it's alright. Don't worry about it.

[00:39] RESEARCHER:

I'll start by introducing myself and we'll go from there. My name is [redacted]. I'm from the [redacted]. I do research on the subject of quality assurance in software process development. I'm doing research currently on the topic of quality assurance practices in Agile methods and with specific focus on the Scrum methods. Basically, why we doing interviews because we like to see in practice how do people implement this quality assurance practices. How do they manage to assure quality in software using Agile methods? So, the belief and the perception of people are very important to us because we believe that knowledge stored in people's heads and they are practicing the methods itself. So, that's why we are doing interviews. If you don't have any questions, we can proceed

[01:53] PARTICIPANT 16:

My questions are more around, if we are spending next 1 hour together so I should know exactly where are you at and what exactly would you like to have out of me.

[02:17] RESEARCHER:

So, basically what I like to get out of you is just your knowledge. What you have been doing in practice. So, you worked as a Scrum master and you worked as a project manager, so you have been in a position where you have overseen the process from end-to-end and you supervised people. So, what I need is your knowledge, what's been happening in these instances of projects where you worked as a Scrum master for example and the level of the detail if you manage to give me examples that would be great

[03:00] PARTICIPANT 16:

Alright so. Are you recording this?

[03:07] RESEARCHER:

Yes. Are you OK with it?

[03:10] PARTICIPANT 16:

But I would stay anonymous?

[03:12] RESEARCHER:

Yes. Your participation is completely anonymous and the examples that you will share with us, the companies, the names you share with us will remain anonymous.

[03:24] PARTICIPANT 16:

I won't share companies' names. I would not share that.

[03:28] RESEARCHER:

Yes, that's up to you.

[03:29] PARTICIPANT 16:

OK, I will tell you how you can benefit from me.

[03:32] RESEARCHER:

Yes.

[03:35] PARTICIPANT 16:

I will get into the benefits stage later. I'm quickly making notes so that I cover, I have my own parts that I got. Before that, I would like to know from you, from your past experiences have you had discussions around TDD or BDD? Are you looking for those kind of examples? Or you're looking for more round how the work is shaped, how the process flows? Are you trying to understand the methodologies of the process flow? What exactly?

[04:24] RESEARCHER:

Correct and how does quality fit into those flows. Before we do that, can we agree on what is quality?

[04:31] PARTICIPANT 16:

Quality can be defined in many ways. The definition may differ from person to person. But, there should be some standards. So quality for me is a product meets customer requirements and free of defects.

Well, so I will give you my philosophy, how I approach it and I will also give you how I implemented it and I will also give you a bigger picture. I saw your profile, you're also coming from a business analyst and you're also coming from I.T. delivery kind of background.

[04:58] RESEARCHER:

Correct, yes.

[05:00] PARTICIPANT 16:

So, when we say assurance, there are many factors when people speak about Scrum and Agile and project management, there is one most important thing and there are very smart people who are into I.T. deliveries and they understand the contract. So, how the people have been contracted, whether it has been outsourced, whether their employees are contractual or full-time. There are many factors that come into play and quality assurance is also part of it. So, I have seen things done in different ways. I have seen very skilled QA teams, functionally very strong because my belief is that actually a good QA team writes its test cases on the business requirements, not on the development user stories.

[06:10] PARTICIPANT 16:

So, if you are a businessperson and you say that I will need to have a website to be able to sell my product, the test cases should be written on the user story, on what the user is looking. But one of the mistakes with the QA has been doing is writing...you know, to be honest, we have not been able to get rid of the waterfall methodology, but that is one of the challenges, which most of the companies have worldwide. They have not been able to get rid of the waterfall methodology. So even if we are implementing Scrum, Scrum says that we should have high skilled workers in cross-functional teams so if you understand this bit, a cross-functional team clearly says that well, if you're having an eight member team, four

people that build something, four people that test something. The other four when they develop something, the other four they test it. So that kind of approach would be much more productive, that is what Scrum says. But again, as I mentioned we might be calling things Agile.

[07:49] PARTICIPANT 16:

Well, I have worked for big companies in the UK. Very big companies and companies that are very Agile and stuff like that. But the problem is the silos in the departments and the ways of working in the departments, they don't really allow the teams to work in a very, very Agile way. That is one of the challenges. At the same time, I'm connected with the industry experts, so I will say there are some small companies, which have really adopted to the Agile ways of working. And they are getting the maximum benefit of it, but what is happening with the bigger organizations is, like someone at the CEO level or the CTO level, or the C-level people, they make the noise that we have to go Agile. It puts us on this flyer and all the small departments, everything under the umbrella starts using the word Agile without really understanding it. Agile calls for decentralization of power or rather it says that empower the developers, but the heads of department, they are not ready to let go of the power. So, Agile is truly not ever implemented. Saying all this, I'm pretty sure you must have heard these stories with so many people that you talking to. What is your take on it? Are there other stories similar all over the world or is UK...?

[09:33] RESEARCHER:

Still doing interviews. But I'm happy to share with you the result at the end.

[09:40] PARTICIPANT 16:

Thank you. That's perfectly fine.

[09:42] RESEARCHER:

Yeah, but I have personally experienced the same thing.

[09:49] PARTICIPANT 16:

So, if you speak to people Researcher, this is the truth. We are the people at ground level, we are the people who get the work done. But again, I will call from my experience a lot depends on... well, one of the projects I was doing as a project manager or as a Delivery Manager. There was no push to be Agile, but it was a very high pressure, high risk compliance project. There was millions of dollars at stake, it was a merger and acquisition project that I was doing, and it was crucial for the organization having a C-level visibility. So, when I was doing the project, I was a project manager but I will tell you the benefits which I took from that project is just by implementing Agile processes, like when we say I was just going through the Scrum. I would rather not say Scrum processes, that would be wrong, the Scrum framework would be the right statement. What I mean is if you really look at Scrum, I was doing my own research and I was writing something so if you really look at Scrum, it

speaks about empiricism. Empiricism says that we learn things from experience so smart project managers, what they are really good at doing is that they will try to break up the entire product backlog as a product roadmap.

[11:44] PARTICIPANT 16:

They will start delivering value very soon and they will start organizing the teams' behavior in such a way that, giving the example of the merger and acquisition project, is that I was successfully able to break down the big project into small iterations, have the testing team get involved with the business analyst from day one so that the testers get a complete knowledge of what is the kind of product that the organization is looking for. So, it was more around transparency, everyone has got the entire team including the developers, the waterfall teams. Again, the problem is the structure. There is a developer, there is a tech lead, there is application lead, there is design team, this is not beneficial to the product. Again, in this hierarchy like the developer doesn't get view of the requirements and stuff like that. So, there were many challenges but the best way for it to be done was that have the tester involved with the requirements from day one.

[13:06] PARTICIPANT 16:

So, what happened is the tester has got complete knowledge. The other thing I would like to call out is around the skill set of the testers. That is very important. Usually again this is something which people love to keep it under the carpet is that testers are usually having lower skill sets than a developer. So, what happens is, for example, if you're a tester and I'm a developer and I have developed something but if you don't have the knowledge of how to test it, you cannot challenge me back. And, if you can't challenge me back, you won't have a robust project. So, the tester is at the mercy of the developers saying, OK, you do it this way, it will pass. That is one of the challenges where we miss on creating a robust product because the testers skill set are not seriously looked upon as we look at a developer's skill set. That is a gap we need to bridge. When we mention testing in iterations, again all boils down to the tester's knowledge. If a tester is knowledgeable of the business requirements, he can participate in the product increment planning or this sprint planning saying that these bits can be developed. Let's divide the project into three iterations, A B and C. So, A can be tested after the first sprint. B can be tested after the second sprint and C can be tested after the third sprint. The testers knowledge of the requirements helps reducing the bugs significantly and make the testing much more efficient.

[14:56] PARTICIPANT 16:

So, this can really, really be bridged if we have got highly skilled testers within the team. It's very easy, it's not a big challenge. It's just that the people who are leading the departments, they always consider the testing team to be like second cousins. Just take part. That really doesn't work. If you ask me from an Agile perspective what testing is happening, I would say that nothing much has changed other than what happened in waterfall. Reason being we are not taking the skills of the tester seriously. I was recently speaking to someone and that might be my next project, they have really scaled Agile. But, when I was looking into their testing teams, it is again the same mistake. It is again, an outsourced team, low skilled team seated elsewhere who will be testing the developer user stories. I wrote a quote to do this.

No, I don't think a tester is supposed to do that. He should be testing requirements, what exactly the business is looking for.

[16:28] PARTICIPANT 16:

There has been few times which again are coming from smaller organizations, I would say. Smaller organizations, lean organizations, there have been these positive things when I see that they don't hire testers. They go for full-stack developers and these full-stack developers are like pretty good. There's something called XP, where one programmer is developing, the other one is testing and vice-a-versa. I think that is pretty good although I have not seen it happening in big organizations. I think XP solves a couple of problems. I also like the Spotify model of doing things. You might be aware of it. But I have not worked on Spotify as of now.

[17:35] RESEARCHER:

Sorry, what is this Spotify model? I'm not familiar with it.

[17:41] PARTICIPANT 16:

OK, it's one of the Agile methods. It's a pretty mature model, I kind of like it. This is something which is pretty good. So, these elements that we discussed until now, I'm quickly giving a recap because I went all over the world to explain it to you.

[18:13] RESEARCHER:

Yeah, so I do have some follow up questions, but a recap is good.

[18:18] PARTICIPANT 16:

The recap is, first a tester's skill set. So, if a tester has the right skill set, he will be able to be a part of the requirements discussion and the design discussions and understand. Because if the design not right, the performance might not be right. Testers should be good enough to understand that the performance is not good reason being it has not been designed correctly. Having early involvement of a tester analyst in the product and when I say tester analyst, there is something about Scrum, which is a risk I would say. What Scrum calls is a lovely way of delivering a product that working in a cross-functional teams and stuff like that, but there is one risk which I always call out when I am working with Scrum teams, is that we need to have a test manager who can have this complete view. Complete view is something like when the Scrum team is working, it sometimes puts the blinkers on. It's going this direction, I have to do this product. A robust product.

[19:44] PARTICIPANT 16:

But a complete view who looks into if there is any data breach, who looks into compliance, those little bits. So, we also need a seriously qualified, highly skilled test manager. So, I got a little bit of testing experience as well, though I was not very close but overlooking things.

So, as a Scrum master I used to play that role of that complete view of end-to-end because the product team is always about product and the features. Who looks into the bigger views on compliance and regulatory and stuff like that? So, coming back, a highly skilled test manager or test analyst might be point five time in the Scrum to give an overview. He or she should be highly skilled, highly empowered. Empowered, reason being I would love to see a test guy in the team who challenges developers at the same level, rather than being guided by the developer. That's how I would like to look at things.

[21:05] PARTICIPANT 16:

Again, we should get rid of that waterfall mentality rather we should get into being an Agile kind of thing when it comes to testing. One of things, is the process flow for testing can be a mismatch. If for example you're a tester and you found ten bugs, I'm a developer who is already working on five things, and if you come and say these 10 bugs need to be resolved ASAP. So, one of the things which I really feel sad for these big companies and this is one of the achievements from the last organization that I was working in, is setting up a company process around testing in Jira. Big organizations, I've seen in the UK, they have Jira but they not really able to have that complete integration of a user story. Every user story has its own journey. And I strongly believe in capturing this complete journey in Jira. And what I have seen is the failure of organizations around been able to capture that user journey into Jira. I have not worked in Trello or any such tool. I'm good at Jira. And I'm a strong believer in transparency. Jira gives everyone a transparent view as to what's happening with the user stories. Jira gives you a complete workflow around how to manage your testing. And again, I will repeat, I have not worked on Trello so no idea. But I strongly believe if Jira can be used appropriately, we can have a complete traceability matrix within the user story.

[23:19] PARTICIPANT 16:

The way we're doing this sprint and the product backlog management and all the bits. It all can be done within the capacity of the team. So that was what I did in the last project and that really adds to my confidence. I'm creating a course on Udemy on this thing, so once it's there, I will be sharing it. That's what I am working on next. So, that's the recap plus the solutioning bit. If you really want to do QA in the way that Scrum defines it. Let's be very clear here. Scrum is meant for product development. The birth of Agile, if you remember, in 2001 when they met in Utah. Skiing was 2001, I believe.

[24:25] RESEARCHER:

2001, yeah.

[24:27] PARTICIPANT 16:

2001 yes. The term that we used is organizational anarchists. The 17 developers. The other rebels. The other rebels being there are two kinds of mentality or there are two kinds of thinking processes. I will just give you a quote from my last manager I was reporting to. I was doing a very critical project for the organization and they will not able to move forward because of some data issues. We had GDPR and stuff like that coming in last year in Europe.

[25:05] RESEARCHER:

Yes, correct.

[25:06] PARTICIPANT 16:

So, what happened is I suggested something in the meeting which solved the issue. And it removed the impediment, it made the project move ahead and my boss came, and he said, you're a project manager, you're not here to solve the problems you are here to raise risks. And at that time, I felt bad but as I started maturing what I could understand is, yes, it is an organizational structure which is very much required. Sprints speaks about delivering quick value to the customer, that is also required. But there is a middle part. So, a successful project manager or an Agile or Scrum master has to get that balance right. So, I think that's about it. I would lastly emphasize that having a good process management workflow in place around the floor of a user story or which maps the user story journey is very important to have successful testing being implemented. There are three things I would say: support from organization, taking testing seriously not just as a tick box, understanding what Agile is from an organizational perspective. The third bit would be getting the right skill set of the testers and lastly having the right process around the user story journey in which QA is a part of. That's how I would sum it up.

[26:58] RESEARCHER:

OK, fantastic. Do you still have some time because I do have questions?

[27:05] PARTICIPANT 16:

I kept an hour free for you.

[27:07] RESEARCHER:

OK, fantastic. You talked about the journey of a user story and embedding QA in that journey. How do organizations do that now?

[27:22] PARTICIPANT 16:

Of all the organizations that I have seen, they don't have any central policies. A central live stream as to how they would like to do it. It's left up to the individual project managers or program managers.

[27:44] RESEARCHER:

As a project manager, how did you decide to do it? Can you give me an example?

[27:52] PARTICIPANT 16:

The example is I used Jira in which I created a workflow. Again, you won't be surprised. Your first question was how do organizations do it, when I say that the organizations I've worked with they didn't have a central way of doing it. What I'm speaking about is the delivery. Organizations they might be thinking at the C-level that we have a framework to do it, but it doesn't work at the ground level. How I did it is, I started capturing the entire user journey in Jira. Ensuring that I had a workflow. Get the requirements, it goes into the build, it goes into QA. If QA feels that something is wrong, raise a bug, it goes to the backlog. During sprint planning, if you're in the testing phase, keeping some capacity free. Twenty percent capacity free just in case the bugs come in. There are many ways to do it. But again, these decisions, a scrum master is a facilitator where he can come up with ideas. I can say that guys, is it ok to the product owner, can we just dedicate one complete resource. You know when I say twenty percent, it's like what one resource might be if it's a team of five or six. One resource for bug fixing.

[29:38] PARTICIPANT 16:

At times I've been lucky to have the testing within the dev team. At times I've been unlucky where some of the companies they have got a centralized QA team. Everything has to be stamped by the QA before it goes into production. So, in such scenarios, go with the process. Sometimes I've played smart when I say, played smart, it's something like I got the QA manager for governance. So, he would be looking into governance, testing team I would put within the sprint team. The governance manager or the QA, he would be ticking the box for the organization whereas the testing team, within the team, would be doing the actual testing. I think that's a pretty good way of doing it. But I was speaking like a selfish project manager, more concerned about getting his project over the line.

I don't know how it would work in all the scenarios. In my scenario it did work because it was a critical project, it had C-level visibility, and overall, I had a very good understanding of the organization process. I knew that it would work.

[31:08] RESEARCHER:

So, at what stage does the testing start and what time of testing in this scenario was done?

[31:22] PARTICIPANT 16:

Different projects comes with different flavors. I think the testing starts as soon as the sprint one is delivered. But that is not the ideal way of doing it. Testing should start the moment the requirements are being defined. That is the ideal way of doing it. Reason being the preparation for the testing should start when the requirements are being defined. The test analyst or whoever it is starts engaging with the requirements from day one to have a very clear understanding of what business is looking for. There is something called test-driven development. You write the test case and then you write code for the code to pass. I also think that is a very good approach but again it goes back to the skill set of the tester. In my entire experience, of twelve years plus, I'll be honest, I've just seen one testing team which was really good. The reason being they had been testing that particular product for more

than five or six years. They were really good. Otherwise, many times testing has been an overhead for me, rather than any help.

[32:54] RESEARCHER:

Is that a cultural issue in most organizations that testing is an overhead?

[33:01] PARTICIPANT 16:

I think it's cultural. Partially I would say yes and secondly, if for example, you are an ETL developer, if the market pays you five hundred pounds a day, but if you are an ETL tester, nobody wants to pay you more than three hundred pounds a day or even less. To be very honest, there is no job description called an ETL tester. Let's be honest about it. What happens is that there is an ETL developer and a generic tester would be testing it. So, how could you expect that same level of challenge coming out from that kind of a tester.

So ideally it should be one guy's developing, the other guy is testing, this kind of an approach. So testing is again a cultural thing. Which I've seen is testing is looked down upon. It's like for the less smart people. Does it make sense?

[34:24] RESEARCHER:

No, it doesn't.

[34:27] PARTICIPANT 16:

It doesn't. But that's the truth.

[34:28] RESEARCHER:

But that's the truth, yeah. Another question, I sense that quality overall from you're insight, is waterfalled somehow and that's not a good thing. Is that what you tried to convey? That waterfalling quality is not a good thing and it is happening in these environments.

[35:06] PARTICIPANT 16:

I don't understand the question, sorry.

[35:08] RESEARCHER:

Waterfalling quality is implemented in a waterfall concept for quality in an Agile environment.

[35:23] PARTICIPANT 16:

It's not good, yeah. You're right.

[35:29] RESEARCHER:

Why is it not good?

[35:40] PARTICIPANT 16:

In some cases, it is good. In some cases, it is not. It is more like when we are doing Agile, when we are doing Scrum, so Scrum is like this. The word Scrum says that football we are in together, not many things happening at the same time. Whereas waterfall is a relay race. Development leads to testing, so we are not able to and I need not call out the benefits of Agile. The point I'm trying to make is, the benefits of Agile are lost when we do testing in the waterfall way. Does it make sense?

[36:33] RESEARCHER:

Yes OK. I understand.

[36:35] PARTICIPANT 16:

The benefits of Agile are lost if you're doing that kind of testing.

[36:44] RESEARCHER:

What are these benefits of Agile that you're referring to?

[36:48] PARTICIPANT 16:

Shorter turnaround time for the product. Shorter turnaround time, continuous improvement. These two would be very important. Again, the quality of product which comes out is better. These three things I would say.

[37:20] RESEARCHER:

OK. Let's discuss them. A short turnaround. I've noticed from my own experience, but correct me if I'm wrong. It creates pressures in the team. When it creates pressure in the team, they produce poor code.

[37:42] PARTICIPANT 16:

They do?

[37:45] RESEARCHER:

You agree or you disagree?

[37:48] PARTICIPANT 16:

You're one hundred percent right.

[37:49] RESEARCHER:

How do you overcome that in Agile? How is this a benefit?

[38:00] PARTICIPANT 16:

I couldn't understand your statement.

[38:04] RESEARCHER:

So, what I'm saying is short turnarounds creates pressure in the team.

[38:09] PARTICIPANT 16:

Perfect. So, when I say short turnaround, I don't mean Agile is pushing a sixteen-hour job to be done in eight hours. What was happening when waterfall was in place, from the time you come up with the requirement to the time when that requirement goes out in the market was like three years or four years. In Agile, from the time the requirement comes out to the time the requirement is being fulfilled, it was reduced to one year, six months. This is the turnaround I'm speaking about. The biggest misconception about Agile is that it burns out the resources. That's not true. Reason being when we are doing waterfall, let's look at the waterfall pyramid. You've got time, you've got scoped time and cost.

[39:10] RESEARCHER:

Yes.

[39:12] PARTICIPANT 16:

So, for example, if the scope is changing you have to give more time.

[39:24] RESEARCHER:

Sorry. I lost you. Hello?

[45:13] PARTICIPANT 16:

I'm sorry, Researcher. I dropped off.

[45:15] RESEARCHER:

No, it's OK. We can continue now.

[45:18] PARTICIPANT 16:

There was something I was answering to and I don't remember the question.

[45:23] RESEARCHER:

OK, so we were talking about short turnaround. So, we were discussing that short turnarounds creates pressure on the team and it my opinion, it doesn't produce quality. You were discussing your argument.

[45:41] PARTICIPANT 16:

Yes, so I would say that short turnaround... what exactly do you mean by short turnaround? By short turnaround I don't mean that something that was estimated to be delivered in five weeks, Agile says that you deliver it in three weeks. No, that is not the short turnaround that I was talking about. I was speaking about the short turnaround about the conception of the product to its delivery. I will not agree in saying that Agile causes a burn down of resources. No, I don't think that's true. In fact, I would go otherwise saying that in waterfall, what used to happen is that a development team is, and this is again why I feel for the testing people, if a project is to be delivered in for example, three months, and it is agreed with everyone that development will happen for one month, and testing will happen for one month and the other month would be more around preparation and implementation. But usually happens is that development takes more than thirty days. It does into forty-five days and the poor testing guys have to squeeze in their testing schedule in fifteen days. So that is where you get a low-quality product and a burndown of resources. That I have seen always, always, always happening. So, that's my view on it.

So, short turnaround doesn't mean something that is supposed to be estimated to be done in five weeks then someone tells you to do it in three weeks. No. It's rather from the day the project is kicked off to the delivery day.

[47:36] RESEARCHER:

The second benefit you talked about is continuous improvement. Can you discuss that?

[47:46] PARTICIPANT 16:

Continuous improvement over where? There are many ways to do continuous improvement. Which specific bit are you talking about?

[48:04] RESEARCHER:

The continuous improvement of the process, I guess. Isn't it?

[48:10] PARTICIPANT 16:

Agile speaks of empiricism. So, empiricism says that, it might be as simple as that, where two or three people are working together, and something can be done better if we approach something in a different way. We adapt to it. Adaptation is one of the pillars of empiricism. So that is continuous improvement in process. It is as simple as that. And we always do our sprint retrospectives to go ahead with it.

[48:58] RESEARCHER:

The last and the most important things you mentioned as a benefit of Agile is the quality of the product. So, how does Agile or Scrum deliver quality in your opinion or from your own experience?

[49:15] PARTICIPANT 16:

I will quickly refer to the Scrum guide because somewhere it says very clearly that business should work in collaboration with development teams. After this discussion when you look into the Agile principles, you'll see it there. It works with the development team, when business starts working with the development team in the early stages. Let's take an example, a product is going out in....today is April and something is going out to the customers in December. So, in case of waterfall, what would happen is the first time the customer gets to see the product would be during the user acceptance testing. That would be sometime in November or October at the earliest. If he doesn’t like the product, certain changes. The worst thing for a project manager to happen is that in UAT a bug is found. Reason being is he will have to take everything back. It could be design, it could be requirements, you don't know where it goes. It's a real pain. What happens is when you give business the early visibility which Agile says. When I start involving them from April and May and June and July, August, September, then they see the product in October, there are no major surprises. So, if there are no major surprises in December when the product goes out, it will be a product of much better quality in the Agile world.

[51:34] RESEARCHER:

OK, fantastic. I have a last question. It's rather a provocative one.

[51:43] PARTICIPANT 16:

It's alright.

[51:48] RESEARCHER:

What do you think of this statement: Agile produces poor software?

51:56 PARTICIPANT 16:

No, I don't think that this right. I could put it this way, people who try to fake Agile, they might produce a bad quality product. But I don't agree to it. In fact, I would rather say, the only way, again a very provocative statement, the only way for a project to succeed is how smartly the project manager takes in Agile ways of working. I have seen very few waterfall projects succeed in a real successful way. The reason could be, I'm a new-age project manager, I'm thirty-five, I've been doing project management for the last five years. I'm a PRINCE2 but I love doing Scrum, I love doing Agile. Reason being it gives me a raw sense of satisfaction.

[53:16] RESEARCHER:

How do you get this sense of satisfaction?

[53:22] PARTICIPANT 16:

Again, it's more of a human psychology. Success gives you satisfaction. When your project fails, you don't feel satisfied with yourself. You always got something to complain about. If it succeeds, you get a sense of satisfaction. As simple as that.

[53:46] RESEARCHER:

And.. and..

[53:48] PARTICIPANT 16:

Again, my apologies. One more bit, Agile speaks of value-driven deliveries. The value that you're adding to the organization or the project, or your team, is also a source of satisfaction.

[54:08] RESEARCHER:

So, you've been feeling that you delivering value?

[54:12] PARTICIPANT 16:

Yeah.

[54:16] RESEARCHER:

Any final thoughts before we conclude?

[54:20] PARTICIPANT 16:

Not on the official side of things.

[54:24] RESEARCHER:

OK, I've stopped the recording and we can chat. Thank you.