01:25 RESEARCHER:

Good morning, PARTICIPANT 33. How are you this morning? Can you hear me? Good morning. How are you this morning?

01:42 PARTICIPANT 33:

I'm great. RESEARCHER. How are you doing?

01:44 RESEARCHER:

I'm doing very well. Thank you. Just give me a second. I'm just going to... ah this is better. Okay. It looks sunny and beautiful behind you.

02:00 PARTICIPANT 33:

Yes, it's South Africa.

02:03 RESEARCHER:

Lucky you. So, you on holidays?

02:03 PARTICIPANT 33:

Yes, visiting my family. But as you know I live and work in the UK. You in Denmark, right?

02:05 RESEARCHER:

In Copenhagen, Denmark.

02:07 PARTICIPANT 33:

Oh, lovely. Lovely.

02:10 RESEARCHER:

So, we have a sunny day. Sunny Denmark, Sunny. So, it's not like South Africa. Sunny but yeah, it's nice. This is summer for us. Yeah.

02:23 PARTICIPANT 33:

Okay, great.

02:24 RESEARCHER:

I'd like to start by thanking you for, for your time and participating in the interview. I appreciate.

02:32 PARTICIPANT 33:

My pleasure.

02:33 RESEARCHER:

Yeah. So, I'm just going to start by introducing myself, telling you what I do and the process of the interview how we're going to go about it. Okay. So, yeah, my name is [Deleted to preserve the participant anonymity]

02:55 PARTICIPANT 33:

Yeah. [Deleted to preserve the participant anonymity]

02:55 RESEARCHER:

[Deleted to preserve the participant anonymity] But I lived in Denmark for a while. And I work for the I.T. University of Copenhagen, one of the universities here. And I do research on the topic of software teams, and how they managed to achieve quality, what helps them to achieve software quality. Currently, I'm running this Scrum project, I'm specifically looking how the Scrum method helps software team to achieve better quality, whether it makes a difference or not. So, we do interviews, because we like to understand people perspectives, people who work in Scrum teams, and we'd like to capture that experience in these interviews. And we analyze the interviews, and we make, or we draw some conclusions, which we try to report in some form or another and disseminate the knowledge we captured. So, do you have any questions for me regarding what I'm doing before we start?

04:11 PARTICIPANT 33:

Firstly, it sounds very interesting. And I think you've picked up on an area that shows enormous potential. And this is coming from a perspective, you know, when I started in industry, which was in ninety-one, Waterfall and, you know, the traditional project PMP type approach was prevalent, and you know, so I saw Agile, the benefits of Agile, particularly started in the early two thousands. And, and I think that it shows a lot of promise, obviously, there's no panacea in terms of lifecycle in software development. But I think compared to Waterfall, it has significant potential, and I think you're in an interesting area, also, what you guys are doing, do you disseminate this information to industry?

05:07 RESEARCHER:

Yes, we do. We do sometimes present in the industry. And we also send the report to our funders and things like that. But for our participants, those who wants to be to be part of the dissemination, we send them a copy of the report, and we seek for the feedback. So, if you want a copy of the report or the paper, I'm happy to send it to you once it's become available.

05:40 PARTICIPANT 33:

That would be awesome. Thank you.

05:43 RESEARCHER:

Yeah, I'll do that. So, I'm just going to explain briefly how the interview works. And we can start. So, I have a set of questions I'd like to go through. But it is a little bit fluid. At any stage, you want to add an example, or you want to add something, please feel free to do so. Okay.

06:05 PARTICIPANT 33:

Will do so.

06:06 RESEARCHER:

Okay. Thank you. We'll start with a presentation. Just a brief presentation. Basically, what is your experience and what you've been doing for the last thirty years? And we'll start with the questions.

06:22 PARTICIPANT 33:

Okay, so basically, I just need to give you a brief overview of what I've been busy with the last thirty years. Well, my academic background is computer science. I graduated in ninety-one. And then I worked for, you obviously, won't be familiar with the local or South African companies. So, I work for a large insurance company in South Africa that I got my bursary from. So yeah, so I started in the industry, then and then I don't want to give you too much detail but basically within the actuarial space, so to do working on their products being in an insurance company with the actuarial team. When I moved back to UK, I'll skip that, I worked for [Deleted to preserve the participant anonymity] for a while. Yeah, some interesting projects, particularly in Hungary, we did some work there. And then I decided to go into contract teams in the late nineties. In the nineties, it was mainly clients. And so, I've always been in software development. That's my area. So. So at [Deleted to preserve the participant anonymity] I worked in banking, a lot of banking stuff. For, as I said, some European clients with we did some crypto stuff in Hungary for [Deleted to preserve the participant anonymity] which is based in London, then we also did some domestic work with the banks. And thereafter I did contract, and after looking at a lot of the client server type architecture development, the web came along. Sorry, my dog was just forcing me to rub him.

08:08 RESEARCHER:

No, it's okay, I understand. Don't worry about it.

08:10 PARTICIPANT 33:

Thank you. So, after that I focused on the web. So, my focus has been for the past twenty years plus on what has become known as single page applications. So, and that all started with me, just trying to mimic the client server architecture within the web browser. So, when my contemporaries in the late nineties and early two thousands were doing what is referred to as server-side rendering with ASPs and JSPs and ColdFusion. My preference was to play with single page applications. And I was essentially, you know, long before the name developing pseudo single page application. So that is my, actually my forte. And so, during, from, say, from two thousand to two thousand and ten, when, you know, it wasn't really clear as to which way the web was going, I stuck to my guns with ASPs. And by two thousand and eight, I actually thought I'd made the wrong decision, because I couldn't really see that it was the way I was doing this was gaining significant traction. But I was pleasantly surprised post two thousand and ten, when with the proliferation of the JavaScript frameworks, and you

know, later on, Angular and React came to the fore, but Bitview lagging somewhat behind. And, yeah, it was almost sort of a validation that my sixth sense my gut feels for what, for the way that a web application ought to be developed, became mainstream. So more recently, my focus is still on single page applications. I work with financial institutions banks; I spent the last year at [Deleted to preserve the participant anonymity] which we have one of the big five in the UK. I would say banking, banking logistics, we have a predominantly financial in the financial area, developing back office single page applications for enterprises.

PART 2

00:02 RESEARCHER:

Hello, hello.

00:47 RESEARCHER:

Good morning, PARTICIPANT 33. How are you this morning?

00:50 PARTICIPANT 33:

Alright, RESEARCHER. And you?

00:51 RESEARCHER:

I'm doing very well. Can you put your audio a little bit up? Just a little bit?

00:59 PARTICIPANT 33:

Okay, I shall try.

01:01 RESEARCHER:

Yeah, this is perfect. Yeah, this is perfect. Thank you. I think we've done the introduction last time. We will skip the introduction, there is no need to repeat ourselves. And thank you again, for making time today. I'll start with the questions. And I'll explain first the interview, I have a set of questions I'd like to go through. But it is fluid and flexible. So, feel free to add any things you feel you want to add or example you want to give me. So, at any stage. I'll start with the first question. What do you think of Agile? I mean, you have seen Waterfall and you have seen Agile, what do you think of it? And, specifically, I want to understand your perspective. What did it bring new to software development?

02:02 PARTICIPANT 33:

Okay, so it's actually multifaceted. But if I understand the question correctly, you're asking, in contrast to Waterfall, what new that Agile brings to the software development. I don't want to use word lifecycle, but more to the software development discipline.

02:23 RESEARCHER:

Yeah. What is different on the team level, especially how does it help delivering software better than Waterfall for example?

02:38 PARTICIPANT 33:

Okay, so it's actually multifaceted. Let's start with probably one of the secondary aspects, but I think one, it's very important to me, because in the Waterfall world, so a lot of my work back in the nineties, was done with what was become now known as [Deleted to preserve the participant anonymity]. Back then it was the management consulting wing of [Deleted to preserve the participant anonymity], the auditing firm, and they were called [Deleted to preserve the participant anonymity] And so, they're very professional. And, in fact, my mentor when I was young, probably your age was from [Deleted to preserve the participant anonymity] And the approach was based on PM BOOK, are you familiar with PM BOOK?

03:26 RESEARCHER:

Yes.

03:28 PARTICIPANT 33:

Okay, great, great. But they had an adaptation I would call method one, method slash one. I'm not sure if you're familiar with that approach. And what I saw, and it actually was quite blatant, probably the best project manager I've seen in my thirty years in industry was from, I'll just call it [Deleted to preserve the participant anonymity], because that's what they call nowadays. It was from [Deleted to preserve the participant anonymity]. And there was a heavy dependency on her. Carol would be and I was in my mid to late twenties. And so, I could, you know, typically, it was normal for programmers like myself to be at the office at seven o'clock in the evening, you know, even eight o'clock or even 10 o'clock, but at typically at [Deleted to preserve the participant anonymity], we didn't have to, it wasn't [Deleted to preserve the participant anonymity] So, but Carol would be there, and Carol was well into her thirties, I would say about thirty-five, give or take a few years. And Carol would be there because of the heavy reliance on her as the project manager of the various streams within the overall program. And I think Agile, what Agile brought was that dependence on that core person, namely the project manager, and many times the manager, not in Carol's case necessarily, but many times over the years, you know, be it on projects at [Deleted to preserve the participant anonymity] or whenever else at the numerous other places I've experienced it. Many times, the project manager is not technical. So, they are they have some, you know, maybe they've done some engineering background academically or, you know, maybe something in the arts or something, but they, you know, have a formal qualification typically, but they may not have an information technology or computer science background, or software engineering background.

05:29 PARTICIPANT 33:

So, this person has to firstly try to understand the concepts, and then put the project plans together and stuff like that. So, and when you're trying to report to that person, you need to translate it into language they can relate to. So, given all those ancillaries and peripheral concerns, there's a heavy reliance on that type, on that person in that core role. Because it's all it's almost as if the weight of the project is, is resting on their shoulders. And the developers who are actually doing the work are secondary, you know, and I think Agile has turned that on its head, it's almost like socialism, or, you know, what Marx and Lenin and those guys did, in the late eighteen hundreds, early nineteen hundreds, I think somewhere around there, so it was almost like a revolution. When I first saw it, I thought I associated with George Orwell satire Animal Farm. I don't know if you familiar with it.

06:33 RESEARCHER:

Yes, yes.

06:33 PARTICIPANT 33:

So, I was quite cynical, initially. But eventually, I started to see the merits, and not because of any, you know, formal education, just because in practice in it, I thought, No, this is actually good, because now it empowers the workers, the people on the floor, who actually develop the deliverable. And you know, what they doing, you know, and how this ought to be done. So, I think that's the one thing. So, in summary, based on what I've just said, over the past couple of minutes, it's turns the model, the responsibility model on its head, where it empowers the software developers and gives them more responsibility in terms of achieving the deliverable. And what that lessens the burden on sort of the key management person overseeing the project, right? I don't want to associate a name with such a person.

07:36 RESEARCHER:

So, it becomes a teamwork, and it empowers the individual within the teams. Yeah.

07:42 PARTICIPANT 33:

Yes, yes. Especially the people that actually do the work in terms of, you know, maybe doing the analysis and the testing and the software development. But actually, whereas as the management person has less of a responsibility now. Sorry, again, I've got a dog next to me.

07:58 RESEARCHER:

No, it's fine. I like the dogs. I like dogs. I love dogs. It's fine. So, you're using Scrum, right? Or in the past, you're mostly used Scrum, right?

08:11 PARTICIPANT 33:

I think if I mean, yeah, the term Kanban being thrown around. [Deleted to preserve the participant anonymity] has a very strange idea of what Agile is, I mean, I have the term Agile master. But you must remember that they make no association between Java and JavaScript. So, I read a lot into that. And it's shocking for a big brand like [Deleted to preserve the participant anonymity]. And it's not [Deleted to preserve the participant anonymity] UK. But because the projects we work on are, I mean, the guys are going up and down to Germany. So, they have teams all over the world. So, I think they have a very weird idea of what Scrum is, or Agile is, so they have their own terminology, but it's essentially Agile slash Scrum. Yes.

08:59 RESEARCHER:

Can you take me through the process? How do you implement it? And how does it work in general, from the current and past experiences?

09:08 PARTICIPANT 33:

So, essentially, you know, we have what they call ceremonies, which is the daily, so, okay, so it's a little bit we take a step back, and I'm sorry, again, I'm trying to focus on my dog who's demanding my attention. So essentially, the works broken up into chunks called sprints. And in each of the sprints, you define an incremental deliverable, the deliverable and the priority of that as determined by the business, so the team, a team actually doing the work associated with a representative of the business referred to as the product owner, typically, then, when you start a sprint, you, okay, the product owner creates an artifact called a product backlog, which is basically sort of a granular breakdown of the work needs to be that needs to be done with associated priorities. And when you start to sprint, you select items off that backlog based on the priority determined by the product owner, and also the capacity for seen by the team. And then you tackle that within that that sprint, which is typically a two-week period. And mostly in my experience, I've seen two-week periods. When that sprint is over, you have a retrospective which you have a look at and you say, okay, what have we done reasonably well, what could be improved on? And during the sprint itself, you have your typically daily stand up. Although I've noticed that, for example, on my previous project, at [Deleted to preserve the participant anonymity] we had it every second day. One of the tech leads will say that it's actually not constructive to be saying the same thing every day. So, it made sense. We were all working remotely. The project started during lockdown. June last year, and yeah, so he had a point. So, we moved it to every second day. But on most projects, you have your daily standup where your team members basically say what they've done the previous day, and what they plan to do today. And whatever issues need to be tackled, works outstanding for them. At the end of our sprints, we also do typically some kind of presentation to the product owner or the product owner with certain business users. And that's normally coincides with sometime around the retrospective, then yeah, that's essentially how it works.

12:10 RESEARCHER:

Yeah, I think that's a good description of Scrum, thank you. Because we will be talking about quality and how Scrum helps the teams to achieve better quality, so I think we need to define it to make sure that we are on the same page. And we understand what we mean by it. So how do you define software quality in the context of software development?

12:37 PARTICIPANT 33:

Whenever somebody talks about quality one thought, so almost like a primal of thought, jumps to mind for me, and this is because my area of interest is theoretical computer science. And I remember reading this years ago, you cannot test correctness into a program. So, I just want to put that out there. But I think quality can best in my mind, and this is very, very, this is a program, and I'm really on shaky ground. But in my mind, does the artifact or the product you developing achieve the required the desired requirements, both functional and non-functional requirements expected by the target users?

13:35 RESEARCHER:

So fit for purpose is one aspect of it. Yeah. You also mentioned you cannot test the correctness of a program. Do you mean, fit for purpose is just one aspect of it? And how about the quality of the code, the design, etcetera? Do you consider that another facet of quality?

14:03 PARTICIPANT 33:

Well, it's definitely, see I don't know generally how the wider community defines quality. And this is educational for me. So, I'm paying attention at this point. So, but as a software developer, we know that even if your business users aren't interested in elegance in code or, you know, progressive design patterns, and you know, all that stuff. I think it's important to have something that is reasonably elegant from a maintainability and extensibility perspective. So, if we then the question boils down to if we consider maintainability and extensibility aspects that determine quality or improved quality, then yes, I would say that it's true.

14:53 RESEARCHER:

Okay. Yes, fantastic.

14:54 PARTICIPANT 33:

If that makes sense.

14:54 RESEARCHER:

Yeah, we understand what we mean by quality, and I'd like to move to the next segment of the interview, which is basically how Scrum helps. In your team, what do you do to assure quality in your software development using Scrum?

15:15 PARTICIPANT 33:

Well, I think a Scrum, or TDD lends itself well to the Scrum approach. So, for me as a developer, and now, you know, I'm sort of on the backfoot, because you're really challenging me with these questions. And that's good.

15:33 RESEARCHER:

But a more challenging question coming up. So just prepare yourself.

15:39 PARTICIPANT 33:

I'm not as assertive or sure of myself at this point in time, but I would say, I will typically okay, to be honest, for me, if I can have automated regression tests in my deployment pipeline, that to me is I've done at least like seventy, eighty percent of what I need to do to achieve the quality. Obviously, there is an aspect of, you know, getting some kind of feedback from the users and, you know, for them to assert whether we in line with the desired goals or not.

16:17 RESEARCHER:

Okay, great. Do you think Scrum helps achieving quality or better?

16:24 PARTICIPANT 33:

Yes.

16:24 RESEARCHER:

How it does it?

16:27 PARTICIPANT 33:

Because of the continuous feedback from the users, you know, and also encourages things like TDD, you know, because of the iterative approach. And you embrace change. TDD is an essential component of that, you know, if you don't have automated regression tests, it's just a recipe for disaster.

16:47 RESEARCHER:

In our correspondence, you made a comment, which I'd like to follow up on, you mentioned that the accountability of the team as a whole for the delivery of a successful product. What do you mean, could elaborate on this a bit more on this?

17:09 PARTICIPANT 33:

Sure. It's exactly what I was saying about the [Deleted to preserve the participant anonymity] example. Whereas I felt that [Deleted to preserve the participant anonymity], I always come back to her. So, [Deleted to preserve the participant anonymity] is the project manager. So, the project manager always feels as if the burden is on them in order to deliver a quality system, ever. However, what Agile does is takes that responsibility and hands it to the team. Now, because the developers, the testers, the business analysts feel valued, because they know they an instrumental part of delivering this and this, they're not just worker bees. They take a personal responsibility and therefore provide a personal commitment in ensuring that whatever they do is improved quality. At least I'm speaking for myself, but I've seen this overall, within projects over the years. Does that answer your question?

18:10 RESEARCHER:

Yes, yes. I just want another follow up. So, this accountability, as you said, makes every member of the teams wants to improve, is improving, delivering better quality?

18:26 PARTICIPANT 33:

That's a tough question.

18:31 RESEARCHER:

Because you could improve the process and make it more efficient. But does it mean it’s better quality?

19:15 PARTICIPANT 33:

Yes, from experience, I would say so. Okay, if our definition of quality is fit for purpose, then I would say yes. Because Agile encourages, I used the term personal commitment. There is more of a drive of the development people, the guys on the floor to produce something that they personally invested in. And I think that improves that notion of fit for purpose regarding the software that they deliver.

19:46 RESEARCHER:

Okay, fantastic. We will move to more challenging questions. We talked about accountability, and we talked about continuous feedback which they are quality of Agile or Scrum. What other qualities Scrum brings to the table? Other, which they are not software engineering practices like TDD, and testing, etcetera. What does Scrum bring to the teams that help facilitate achieving quality?

20:20 PARTICIPANT 33:

Okay, again, I'm going to labor on that on that aspect of a personal commitment. Because you're breaking down this responsibility, which previously in Waterfall primarily rested on the core person, the project manager, and they were almost seen as the Prima Donna, of the project. So, I think please say the question again. So, I answer the question.

20:47 RESEARCHER:

So how does Agile or Scrum bring to the table for the team to help them achieve in quality?

20:55 PARTICIPANT 33:

Okay, so, yeah, so as opposed to that Prima Donna approach, the personal commitment, I think, of each individual will want them to produce something better. Obviously, you're taking the approach of having iterations with continuous feedback. You know, it's great, where people can get, or the team members can get feedback on the work that produced. And that's, that's awesome. I'm not even sure from answering the question.

21:34 RESEARCHER:

Yeah, you answering it. Yeah, of course, you answering it. And that's really good. Can you give me an example? Do you have an example from your experience?

21:45 PARTICIPANT 33:

Okay, let's just think through projects. The example of how Agile improves the quality of the system with as pertains to people.

22:00 RESEARCHER:

Yes.

22:01 PARTICIPANT 33:

Okay. I think, overall, the example that I would give, you know, typically, Agile encourages a nine to five workday, no sitting there ten o'clock at night, you know, as like, the old days. It encourages it. But I found that when we close to the end of the sprint, the guys are quite willing to put in that extra time without moaning. Because it's not done on a continual basis. So, if they were burnt out, then there would be some kind of issues and they'd be seeing some kind of reluctance. But generally, because Agile encourages an eight-hour workday, I find that they willing to put in that extra effort. And there again, I would say, that also shows the personal commitment to delivering a quality piece of software, because that is the outcome, and they do it with a much more amiable disposition. Whereas in the past, it was

almost as if you're the project manager is the taskmaster and the team leaders and they've been forced to do it, whereas nowadays, that's the dog. Oh, by the way, he watches TV, so you can see. So, if you have a dog there, he's probably going to bark. As I say, you know, he's so demanding.

23:36 RESEARCHER:

Yes, I like attention. Yeah. They love attention.

23:42 PARTICIPANT 33:

Dogs in general, I agree. So, I would say, how Agile would impact the improving of the quality from an individual perspective, I would say personal commitment, that they prepare to, to give, you know, in terms of an example, without feeling compelled to do so. It's almost as if they want to produce something as better as a personal objective. So, putting in that extra effort, staying in and during lunch or whatever, almost becomes a pleasant voluntary activity, whereas in the past, it will be something, ah we have to stay late again tonight in order to do this. As if you have a taskmaster that's driving you to do this. So, I think having a shorter continual workday, in other words, you work in nine to five or you know, eight hours or nine hours a day depending on the project. Which is encouraged by Agile, when the person wants to express a personal commitment to put in extra effort to ensure that a quality deliverable is achieved, that extra effort is not done with a grudging or begrudging attitude. I say that also helped, I think that there's a lot more pleasantness I find on Agile projects, whereas in the nineties, typically you can regularly find, you know, people being overworked, and they felt a sense of strain almost "I must", "I have to do this" and "I have to, I must do this". Whereas nowadays, it's more relaxed. And maybe it's a millennial, the millennials bring an aspect to that as well, I, you know, I'm not really sure. But I find that overall, the projects are much, much more pleasant.

25:34 RESEARCHER:

Yeah. What makes Agile make the project pleasant, is it's the empowerment of the individuals, for example, because of sense, empowerment in there. That's why they put the extra effort. Do you think?

25:51 PARTICIPANT 33:

Well, I think that's definitely, remember, typically, people who work in I.T. are smart, I'm not going to be there's no diplomatic term for that. You know, and I'm not saying in other disciplines and not smarter, I'm not saying, I'm not saying that at all. But I'm saying, you're dealing with an individual on a team, that thinking even if he, he may be sitting quietly in the corner, that mind is working overtime. So, this guy or girl is analyzing the situation around them. And I think Agile, they are probably aware that they should have more say, I'm talking about the Waterfall model, and the deliverables of the project and how we ought to go about achieving the deliverables, than say, somebody who's ignorant, or completely ignorant, but as much less idea of how the software delivery works, or so when I say works, I'm not talking about the process, I'm talking about the code. And you know, the discipline itself. So Agile has given them definitely the right or the opportunity, and facilitated such individuals to feel more empowered, and therefore more valued. And I think that's important. I mean, even in a family context, if you make the child feel valued, their contribution will be more positive. Within the collective group.

27:18 RESEARCHER:

Do you think, you mentioned something very important, their contribution is more positive? Do you think that reflects the overall quality of the software or the product?

27:32 PARTICIPANT 33:

Yes, I do. And the reason for that I say is because they have an incentive, they are incentivized, even if it's not consciously it's subconsciously, they incentivized to have a personal commitment to produce something more because they know that they are a valuable part of what is being produced. And, you know, as a valued part of what is being produced, they want to create something that's of value and better quality, because it's a reflection on them.

28:06 RESEARCHER:

It's again, the accountability things. Yeah. You want to add something, or should I move? Next question. The next question is more detailed. We already cover a bit of it. But I want a more specific if you managed to give me example. How does for example, your Scrum or Agile in general helps find these more bugs?

28:32 PARTICIPANT 33?

More bugs? Oh, that's interesting. That's an interesting question. Does it? You know, I think it's great that we discuss ideas and all the rest and yes, yes. You know, if, if something and I'm just thinking about real experience, now I'm thinking through you know, with guys, I'm just so yes. So, we in discussion, right? Guys give the feedback, and we in discussion the guy, okay, maybe, you know, there's a different approach or, or maybe we should look at Redux saga. Sorry, I'm using technical terms, right, versus maybe we should look at technology A rather than B because we using it here or used it before. And that's great. So that discussion, that type of discussion, within the standup would definitely facilitate a platform for discovering more problems within the system or issues. And also, maybe preemptively prevent bugs from incurring or the waste of time, you know, because somebody instead of working on the assumption that this may be the better approach, a fellow team member can recommend a tried and tested approach that he or she may have used in the past. So, I think that type of discussion definitely facilitates the alleviation of more bugs because having two minds, I think is better than, you know having one. And you know, maybe I think they proposed peer programming.

30:07 RESEARCHER:

Okay, that's a good answer. Thank you very much. I liked it. And I liked the example. We already talked about the next question, which is, how does for example, Scrum help producing high quality code. Yes, this one about the code itself. I know we talked; your definition of quality tends to be more about fit for purpose. But the quality of the code itself, the developer, the drive for the developer to write good code, like you said, maintainable, extensible, elegant, perhaps, make it easy to read for others. That's my understanding for elegant, how does for example, Scrum helps the developer to achieve that level of code?

31:02 PARTICIPANT 33:

Well, you know, again, it's going to be pretty much the same answer, probably sound like a stuck record.

31:09 RESEARCHER:

No, no, it could be the same answer. That's fine. Yeah.

31:13 PARTICIPANT 33:

Because, you know, programmers are more valued, and feel worth something. And therefore, again, they want to make a better contribution. And when somebody feels that way, they're going to, you know, want to develop it. And remember, within a programmer's arsenal, he's got basic skills, and so he's got a handgun, and then he's got an Uzi or whatever, is the top end automatic rifle at this point in time, I have no idea. So, he has different types of weapons in his arsenal. And when he feels compelled as a job just to do this, he'll probably use his revolver or some other handgun. And, but when, let's say if you if he was writing it for his girlfriend, for example, or a girl was writing for her boyfriend, for example, they will put in extra effort because of their personal commitment. So, he would probably pull, or his girlfriend was a programmer, and he was working with her. So, he had this personal motivation, that's where I'm going to, the personal motivation, out of his arsenal of skills for developing software, it probably pulls towards the top end, he probably uses good abstractions in his design patterns. And within his algorithms, rather than just using a blah approach that this has to be done, which is an easier situation to begin with in the Waterfall environment. Because that's sometimes how you felt, you know, I'm not saying all Waterfall projects, but that by no means I'm saying, it's just easier to get to that point where I'm just doing a job. But when you feel valued out of your programming arsenal, you'd want to use something which you can be proud of, to contribute to the product that you're developing.

33:20 RESEARCHER:

Fantastic. I think you already answered my next question, or I'm just going to repeat it. If you want to add something or you have an example, how does Scrum motivate the developer to thrive or to achieve better software quality? So, you already answered that is the motivations and the investment they put in as a team member to produce better software quality or better code, right?

33:51 PARTICIPANT 33:

Correct.

33:52 RESEARCHER:

From your experience, you worked in Waterfall you moved in Agile, or Scrum in general, do you think that Scrum or Agile produce better software quality?

34:05 PARTICIPANT 33:

In general, yes, I do believe that there are elements that you know should be factored in. Because I think there is a gap in in in Scrum slash Agile. And that's something that was covered well in in Rational Unified Process, because Scrum doesn't address how to create the foundational aspects. Scrum doesn't address how to get started. So typically, on an

enterprise, and I'm just talking enterprise software now. In an enterprise system, it's difficult within two weeks to have something visible for the user. And I say this because what you want, and which RUP addressed. Are you familiar with RUP?

34:54 RESEARCHER:

Yes, yes, because I've done my bachelor's degree in the nineties. And it was very popular in the late nineties. It was very popular. Yeah, it's an IBM product. I remember right.

35:06 PARTICIPANT 33:

Rational was bought by IBM.

35:09 RESEARCHER:

Yeah, yeah. At one stage. Yeah. When I started working in the industry, it was popular. And I think IBM at that time already bought it. Yeah.

35:19 PARTICIPANT 33:

Okay. Yeah. So, the Rational Unified Process had this this notion of a candidate architecture, which said, Okay, before you start building anything, and if you built a lot a lot of enterprise systems, you know, that it's not just a website or a CRUD application. I mean, yeah, business. Applications are typically data intensive forms and data and CRUD operations. But in an enterprise application, I would say, were RUP got it right was that it invested time and created a candidate architecture, which means that you have this base platform on which to build. However, Agile says you need to deliver something of value from point from time zero, at the end of that first sprint, the business must see something. I understand the rationale and there's value, most definitely, I'm not negating the approach. But what I'm saying is, I think RUP got it right, by allowing time within the SDLC to create the candidate architecture is essential, I would say, imperative within an enterprise system, for example, think about if you are creating a modern-day event driven back end with Kafka. And, and stuff like that, you know, with streaming that to set that up, is not for kids. Yeah, that's some serious back-end stuff. So, I just don't know how you're going to see value after the first sprint, I would say two sprints for that alone, before the user can actually see the value. And I don't know if I'm babbling, am I answering the question?

36:58 RESEARCHER:

No, no, I had the same experience. I worked in the finance industry for years. And they tend to have large and massive programs, I'm talking about sixty-five million to two hundred million programs. And, Agile become popular in mid two thousands, and we struggle with those projects, because we didn't have the whole picture upfront. So, what's Waterfall or Rationale used to be good at that. You invest a lot of time upfront to build that holistic picture and that candidate architecture, we can call. So as a development team, we missed that. So, the small iterative approach didn't help us, because we were building for two weeks. And in two weeks is not what the client wanted. So, the client liked that product backlog. And that's flexibility.

38:06 PARTICIPANT 33:

And the continuous feedback.

38:06 RESEARCHER:

Yes, and the continuous feedback, but they were given us chunks. And the whole picture was missing. So, the end product didn't meet the whole picture. We keep patching every iteration, we kept patching and patching and patching. And the product didn't end up at good quality unfortunately.

38:29 PARTICIPANT 33:

Sounds like a recipe for a nightmare. Yeah. So, I what I found is and sorry about that, that you had to go through that.

38:36 RESEARCHER:

No. No, it’s learning, ongoing learning. I mean, I think the transition to Agile for this organization was the failure. It's not the theme, I mean, those organization are risk prone organizations. They don't like risks. They are very hierarchical. They like everything to be organized and things to be up front. I think the transition to Agile was a little bit less, was not taught properly. So, they fill in the implementation, we shouldn't blame as our Scrum for it. I think sometimes we should blame the transition to Agile it has caused partly the problem. Yeah.

39:31 PARTICIPANT 33:

Correct, correct. I think many people had teething problems transitioning to Agile and especially if an organization like the large FinTech or financial organizations that are typically risk averse, and I've seen that struggling to adopt Agile. In fact, I have initiatives with a whole approach was just on, I remember working in one of the banks while it's actually like set up investments slash asset management institution partnering with the bank, they had a project which I was on just before two thousand and ten. And they, the goal of that project wasn't to deliver anything. They just gave us something to entertain us. It was to show that Agile works. That was the goal of the project.

40:19 RESEARCHER:

Yeah, yeah, I've been on this project where they have extra money, and they just want to spend it. And because they didn't want to tell the business we didn't spend a lot of money, or the money or things like that I've been on this projects, and they never tell you up front. That's the intent.

40:40 PARTICIPANT 33:

No, they told us.

40:40 RESEARCHER:

They told you? At least as a team, we felt like it was our failure, which it wasn't our failure. They were playing with us. Yeah, I know, those politics of organization can have ethics on project. Yeah. Yeah.

40:57 PARTICIPANT 33:

So, I don't know if I've answered the question, but I think from what I've seen, that issue hasn't been addressed in Agile, so maybe seeing as though you are affiliated with a research organization, maybe you need to initiate something like that. Maybe do some, you know, publish some work. I mean, you can reach out to developers like myself, maybe we'll be more than willing to give input as to how we perceive that issue to be rectified. Yeah, it has to be addressed at some point.

41:31 RESEARCHER:

Yeah. Perhaps it's my next research project.

41:34 PARTICIPANT 33:

Yeah, absolutely. Absolutely. Because it's a real value add, in my mind, you know, I'll put the question out there. In fact, you know, use not just in academic circles, but medium and Cora put the question, how in an Agile world, how are we supposed to address this on enterprise applications, you know, the large enterprise applications, and try to get some idea as to what people are thinking as to how we can approach this because it's a real problem on big projects, which I haven't seen even some of the best development team for, at least locally in South Africa's emerging market, I haven't seen anybody addressing this in a reasonable manner. Anyhow, sorry I digress.

42:16 RESEARCHER:

No, no, I agree with you. And you make me think perhaps it's my next project. We'll see. Do you have any questions for me before we conclude? Because you answered all my question, and I enjoyed the discussion and some very nice and insightful examples. Thanks for that. So, do you have any questions for me?

42:41 PARTICIPANT 33:

Tell me a little bit more about your research?

42:55 RESEARCHER:

I work for the I.T. University of Copenhagen, which is one of the universities here. So, we do research project in various things from theoretical computer science to software engineering. And people like me looked at the team level and how these methods helped the teams to achieve quality. So, and we even have ethnographers, which they look at, at how team works and, and the context of I.T. Yes, I'm affiliated with the university. I don't work for a private organization. I worked for in the industry for years. And I decided to come to academia, because it gives me a lot of freedom to think and a lot of freedom, maybe because I worked for a large organization. I didn't have that. So yeah, I'm happy in academia, it gives me the freedom to think so to answer your question. Yes. And we do a lot of research and in addition to teaching, and yeah, so this is it.

44:19 PARTICIPANT 33:

So, you're a lecturer?

44:19 RESEARCHER:

Yes, yes. I lecture software engineering topics.

44:22 PARTICIPANT 33:

I didn't realize that I thought you worked for an independent research institution affiliated with the university?

44:34 RESEARCHER:

No, I work for the university. Yeah.

44:38 PARTICIPANT 33:

That's great. That's awesome. Can I tell you a secret?

44:40 RESEARCHER:

Yeah.

44:42 PARTICIPANT 33:

When I graduated, many, many, many years ago. My plan was to go into industry for five years and return and just be an academic for the rest of my life. I sadly, I lost. Obviously, money has something to do with it.

44:59 RESEARCHER:

Yeah, the money is better.

45:00 PARTICIPANT 33:

My passion is still academia. In fact, you know, in South Africa, we have a three-year degree called your bachelor's and then we have a fourth year, called the honors. In fact, after twenty-five years, a couple of years ago, I went back to do my honors, you know, and that's after twenty-five years. I'm graduating. I long for it. I yearn for it to be honest. But anyway.

45:24 RESEARCHER:

It's not late. I mean, you still can do a PhD and work in a university. Yes. So yeah, age has nothing to do with it.

45:31 PARTICIPANT 33:

Thank you. Thank you. Anyhow, it was great meeting you. It was really an awesome opportunity.

45:35 RESEARCHER:

So, I'm organizing a workshop with developers like yourself and I will be presenting the result of the research and getting their feedback. If you want to participate, I'm happy to invite you once I'm ready to do that.

45:50 PARTICIPANT 33:

Great. Please do.

45:52 RESEARCHER:

Yeah. Okay. I will get in touch once I organize that, and I'd like to thank you again and wish you a good day.

46:01 PARTICIPANT 33:

You too. It was really great chatting to you. Bye.

46:05 RESEARCHER:

Say bye to the dog.

46:07 PARTICIPANT 33:

Ah, he's sleeping now. Bye.

46:13 RESEARCHER:

Have a good day.

46:13 PARTICIPANT 33:

You too. Bye.