00:34 RESEARCHER:

Good morning, PARTICIPANT 38, how are you?

00:37 PARTICIPANT 38:

Good morning. Can you hear me?

00:39 RESEARCHER:

Yeah, I can hear you very well, how are you this morning?

00:43 PARTICIPANT 38:

I'm good. Thank you. What about you?

00:46 RESEARCHER:

I'm doing really well, it's a beautiful day here in Copenhagen, so I cannot complain.

00:54 PARTICIPANT 38:

Looks like we'll be the same.

00:57 RESEARCHER:

Okay. I'd like to start by thanking you for the opportunity to talk to you and accepting the interview, I really appreciate. And I hope we're going to have an interesting conversation. I start by introducing myself and tell you what they do and explain to you why I'm doing it. And we'll take it from there. My name is [Deleted to preserve the participant anonymity] I'm a researcher at the I.T. University of Copenhagen, one of the universities here in Denmark. And I do my research in software quality, I'm very interested to understand how software teams managed to achieve quality, what do they do in order to achieve software quality. Currently, I'm doing this project to understand how Agile or Scrum in particular, helps software development teams to achieve better software quality. So, we do interviews because we like to understand software teams perspective and experience. And we capture this experience through interviews, and we analyze it, and we draw some conclusion which we try to publish. So, in a nutshell, that's what I do. Do you have any questions for me before we start?

02:29 PARTICIPANT 38:

No, it's clear. Interesting. Happy to help.

02:32 RESEARCHER:

Yes. So, the structure of the interview is I do have questions, which I will ask you. But it's a little bit fluid. If you want to add any things, or you have examples you want to share with us feel free to do so. I start with asking you to introduce yourself telling us what you do what your experience a little bit briefly.

03:01 PARTICIPANT 38:

I work as a developer; I think all my life after graduating from university. So, I started my working enterprise company, it was advertising company, biggest advertising in Russia. And I work there for fifteen years. And about five years ago, I moved to the UK to work for a company. Now, I started working as a consultant software developer for different companies and many projects, many, many clients, many companies. So also in different roles, as a developer as a business owner, as a team lead, as a CTO.

03:58 RESEARCHER:

Yeah, I had a look at your resume is quite interesting. Yeah. And you've been around for a while. So that's a very interesting experience. Thanks for sharing that with us. Just to warm up the conversation. Let's start with this question. What do you think of Agile in general? What is your experience working on Agile software development teams?

04:23 PARTICIPANT 38:

I think right now most of the teams are using Agile in some way so not fully implementing the recommendations, I think. But Agile approach is a major approach right now. What do I think about it? I worked in classical Waterfall projects, and for software developments they worked really not well because people can't, especially in commercial areas, if it's not for scientific or something, which is very stable. So, if you're working in a business environment, everything change in the business, and you need to make changes very quickly. So generally, it's an approach that responds to this frequently changing environment and it's a major approach right now.

05:47 RESEARCHER:

Okay. So, the frequent changes and the fluidity of the methods make it better than Waterfall?

05:55 PARTICIPANT 38:

Yep. Correct.

05:56 RESEARCHER:

Yeah, so, that's good. I'll move to the next question. Yeah, I know that you use Scrum in your team. Can you describe to me the process how do you use it?

06:12 PARTICIPANT 38:

Ah, I work for all sorts of company. I think UK company which sold me to a US software development company inside software. So, we have a product, it's reporting tool, so it's a big project actually. And we have a team about I think eight developers. Every day, we use Scrum in this company. So, we have two weeks sprints. We have daily calls, we have retrospective, and we have demo for the clients at the end of each sprint, so more or less full implementation of Scrum approach in this company.

07:23 RESEARCHER:

So, are you happy with the implementation the way how you approach Scrum in this team, right?

07:29 PARTICIPANT 38:

Yeah, because usually, people say we use Scrum, they just have sprints. And that's all so and nothing else. But I think very important things in Scrum are the demo and retrospective.

07:49 RESEARCHER:

So, you describe the process of Scrum, how does it work. My question now is what do you do to assure quality in the software development process? What tool one, what methods do you have in place in order to achieve better software quality?

08:09 PARTICIPANT 38:

Usual methods in software development, testing. So, we have a guy in the team and as soon as we implement some features, we pass it to review to test. At the end of the sprint, all features developed inside the sprint should be tested.

08:37 RESEARCHER:

So, you do all the testing. What type of testing do you do?

08:42 PARTICIPANT 38:

We don't use unit tests, unit tests, because software itself is quite old. It's very Visual Basic. It's not, not good. Unit tests are not good for Visual Basic actually. We can look for Visual Basic picture. So, we use functional testing. So, this guy testing feature by feature, and sprint, we use end to end testing when you show when you're preparing a presentation for the client to show all functionality implemented within this sprint. This is also some type of testing.

09:39 RESEARCHER:

Do you have code review, for example, do you review each other codes? You're quite senior?

09:47 PARTICIPANT 38:

Yeah, yeah. We use code review.

09:50 RESEARCHER:

Do you have automation, do you use automation tool or linters tool for static analysis for example?

09:59 PARTICIPANT 38:

No, no.

09:59 RESEARCHER:

No? Okay. So, you focus mainly on functional testing to assure quality and code review, right?

10:06 PARTICIPANT 38:

Yeah. Correct.

10:07 RESEARCHER:

Okay fantastic. We will be talking a lot about quality in the next upcoming questions. So, to be precise, we like to define the term. What do you, how do you define software quality in the context of Agile software development?

10:31 PARTICIPANT 38:

Quality is I think it's how developed software meets clients expectations. So, sometimes when a plan meets some big solution, which is, a bug might be not implemented all features he wants but if it solves his current problems, he can treat it as quality software.

11:13 RESEARCHER:

So, is that only the definition of quality, is not to be buggy and to meet the client expectation? How about the code and the design and you are a senior software developer, what do you think of that aspect of quality, the technical aspect of quality?

11:32 PARTICIPANT 38:

Of course, the more careful design you do, the more you will get software with better quality. So, the more you spend time on the product, it will be better of course.

12:02 RESEARCHER:

Okay, so I know that...

12:05 PARTICIPANT 38:

On design stage and development stage on testing stage.

12:11 RESEARCHER:

So, what you're telling me you make sure that every stage of the development you look out for quality right?

12:19 PARTICIPANT 38:

Yeah, right.

12:20 RESEARCHER:

Okay. Now we know what quality is from our perspective. Do you think working in Scrum or Agile environment produce better quality?

12:48 PARTICIPANT 38:

If you are talking about quality is meeting customer expectations, then yes.

12:57 RESEARCHER:

And how does Scrum help to achieve better quality than Waterfall for example? What is good about Scrum that makes it achieved better quality?

13:12 PARTICIPANT 38:
It's iterative and because customer expectations are changing. First of all, the customer can see what you do after each sprint so it's not a Waterfall when customer will see the product actually after the project is finished or maybe after some milestone but still milestone it's a big chunk of time. So, customers will see almost immediately what team do, and can adjust his expectation and feel that quality of what he get is, what is, what he wants. I'll explain because when I worked in [inaudible] we don't use Scrum, not don't but yeah, we don't actually. And when we show what we do to my client, and he tried to do it. He said nothing is working. But maybe not correct, but it’s not working. It's his usual phrase. So that means that we might maybe buy the software, first impressions that quality is awful. But after that, when we try to understand what he mean, we understand that he don't know which functionalities, he doesn't know which functionality we implemented and try. And he, he has his own expectation. And we have software, which might be solve his problems, but in different ways. So constant interaction with customer is very important. Scrum does this.

15:40 RESEARCHER:

So, it's aligned the expectation of the client with the software development team. That's what you're saying?

15:46 PARTICIPANT 38:

Yeah, yeah, absolutely.

15:47 RESEARCHER:

Yeah. So, this alignment give the client the opportunity to give frequent feedback, right?

15:53 PARTICIPANT 38:

Yeah.

15:55 RESEARCHER:

So that's this frequent feedback help you as a software developer to write better code to achieve better quality? How does it do that?

16:07 PARTICIPANT 38:
First of all, you'll get the motivation. So, you'll see is it what you do is needed to the business

to the client and feel yourself useful. And it motivates actually. And also sometimes, we have so called FX Chinese developer, which, who wants to do everything perfectly. And it usually takes a lot of time, lots of resources. And if business said, We don't need that, we need this big solution. It's helps to, I would say to build the quality software in our terms.

17:05 RESEARCHER:

So, this frequent feedback gives you motivations, and subsequently you'll write better software and better quality. Right? That's what you're saying?

17:18 PARTICIPANT 38:

Yeah. Yeah.

17:20 RESEARCHER:

Okay. So that's for the client, the client closeness or the client giving you frequent feedback. You mentioned another quality of Scrum, which is iterative approach to software development, right?

17:38 PARTICIPANT 38:

Yeah.

17:38 RESEARCHER:

How does this iterative approach to software development helps you to, to write better code or to achieve better quality?

17:50 PARTICIPANT 38:

Actually, when you work on some project, if it's a Waterfall project, you just do your work, pass it to further steps in the process to test or somewhere and you will too for example, another project, so you feel your work is finished. But when you work in iterative approach, you know that in next sprint, you will return to this code, you will need to change it and you will try, you're trying to do your code better, better quality.

18:43 RESEARCHER:

Can you give me an example? Because I didn't understand.

18:51 PARTICIPANT 38:

For example, you asking to do some feature. And you know that this, you need to write the codes that can be used in a similar feature in this project later. You have two choices. Make quick changes that solve this immediate problem and build this feature or build more better code that helps you to build a similar feature in future. And you know that next week might or maybe next sprint, you'll get tasked to build this similar feature, and you'll return to this problem again. So, you do not solve this immediate problem, but you solve both the general problem for future sprints.

20:02 RESEARCHER:

Yeah, yeah, I understand. Yeah, that's a good way to say it. That's a good example. So, the fact that there are upcoming sprints, and you always keep in mind that the scalability aspect of your deliverable because there is an upcoming change, which might affect your functionality. That's why I understood right?

20:25 PARTICIPANT 38:

Yeah. Absolutely.

20:26 RESEARCHER:

Yeah. That's a very good example. Thank you. I'll move to the next question. We talked about Scrum facilitate better quality by being iterative and having the client close by and frequent feedback. How about other qualities like retrospective, sprint planning, sprint review? What do you think of those Scrum events or ceremonies? And how do they help?

21:14 PARTICIPANT 38:

For me, particularly, inside software, when a sprint is planned, we have sprint backlog with a task. And every developer can select a task to do. So, there is no direct assigning tasks to developers by some manager or team, everybody decide what he'll do and which task to do. For me, I know this, that I do know this area better, but he knows this area not good. And I do such task better than others. So, I can select tasks that I can do more effectively.

22:14 RESEARCHER:

So, when you say more effectively, is it your write better code?

22:20 PARTICIPANT 38:

Yes, because I know this area, but very likely that I don't do any errors in this code, I take into account all things that need to do. So, I can select tasks when my quality will be better.

22:50 RESEARCHER:

So, when you select the task, you know that you are in control of it technically and even your estimate of the time is better, right. And you have the time to invest in quality, right?

23:06 PARTICIPANT 38:

Yeah, yeah. Correct.

23:09 RESEARCHER:

Do you have a good example for that to share with me?

23:14 PARTICIPANT 38:
For example, if I did similar task in previous if you are, for example, fixing bugs, and I fix this, or bugs in previous sprints, and I see a similar problem in the current sprint, I just take this task and say that I know the reason and I will fix this very quickly.

23:57 RESEARCHER:
Okay. Yeah, yeah, I understand that. Thanks for that. How about that sprint planning? And how about retrospectives? Do you do retrospectives?

24:11 PARTICIPANT 38:
Yes. But actually, in this company, retrospective is just a way to talk about what happened. But there is no lesson learned. So now, I haven't seen any steps planned to do in response of this session when we discussed some problems. So, we discussed what was not good, but it's just stays in mind of each developer should improve it next time. So, no formal conclusions. So that's why I said that Scrum was implemented in different way in different companies.

25:15 RESEARCHER:
Why do you think in your current teams, you don't speak out about how to fix problems and how to improve?

25:25 PARTICIPANT 38:

No, we speak but as I said, there is no conclusion.

25:32 RESEARCHER:

What do you mean, there is no conclusions?

25:34 PARTICIPANT 38:

My opinions are tough to retrospective if you've found some problem. We need to at least write it down and add this to the next sprint. How to fix it.

25:59 RESEARCHER:

Yeah.

26:01 PARTICIPANT 38:

How to fix this problem is the future.

26:04 RESEARCHER:

So, what you mean is the necessary action to improve are not taken right? Do you do sprint review? Or you just do retrospective?

26:19 PARTICIPANT 38:

Retrospective.

26:21 RESEARCHER:

Okay, fantastic. We will get into some deep questions about how Scrum helps. And I need some example would be nice. How does for example, Scrum helps better finding bugs?

26:45 PARTICIPANT 38:

Finding bugs?

26:49 RESEARCHER:

Is finding bugs better in Scrum? And how?

26:56 PARTICIPANT 38:

I wouldn't say it's finding bugs is better in Scrum.

27:05 RESEARCHER:

Why is that?

27:06 PARTICIPANT 38:

Probably when I worked in big company, we had not Waterfall, but some ticketing process. So, you get tickets to do software development, you get this ticket to another team. And each developer for was an individual contributor. So, there is no teams. And each, for example, quality assurance, [inaudible] each person had his own metrics. And there was also a confrontation between different teams, for example, between developers and between quality assurance. So, the quality assurance persons had their own metrics to find some bugs and they checked something saying it was a bug, but it was not a bug. And developer was interested in hiding bugs, to not fail his metrics. In Scrum, we are a team. So, we are working together. And even if developer find some bugs, he raised this issue. And we can fix it immediately, for example, inside the sprint, so I think in that way Scrum helps, as its close teamwork.

29:05 RESEARCHER:
Fantastic. That's a very good example. Thanks, PARTICIPANT 38. I like that example. Very good. I move to the next example. You talked earlier about having the client close to you and having that frequent feedback motivate you to produce high quality. Does Scrum in any other ways help you to produce high quality code?

29:36 PARTICIPANT 38:
There is another side of interaction with a client. It's called sprints, it's during the sprint the team is isolated from new tasks from the client. Because in usual will work when you work on some big task for example, client tries to interrupt you as he thinks of a more urgent task or maybe some bugs and this interruptions will not let you write this task with a good quality. So, you might be switching to another tasks might lead to bugs. And not good performance, I think. So, when you have well defined sprints, you work on this task only, and that's helps you to concentrate and try good quality software.

30:51 RESEARCHER:
That's a very good example. Thank you. I like it very much. What other the qualities of Scrum motivate you or contribute to writing better quality? So, the focus on one sprint in a concise piece of work and what other qualities Scrum in your opinion has that motivates you as a software developer to write good code?

31:21 PARTICIPANT 38:
As I said, it sounds like I said previously, but when you work in Scrum, you work as a team. And you don't want to make team looking bad with the clients, so and you are feeling responsible for writing good quality code.

31:52 RESEARCHER:

That's a good, a very good thing. Yeah. So, it's allowed this dynamic and this cohesion in the team, which make you accountable to write better code, right?

32:04 PARTICIPANT 38:

Yeah. Correct.

32:06 RESEARCHER:

Yeah. Fantastic. I'll move to the next question. We already talked about that. How does your Agile environment motivate you to achieve good quality and you highlighted a few Scrum qualities and how they motivate you? So, I'm not going to ask it again. We discussed that Agile, because you have over thirty years' experience. So, in your experience, how does Agile in general compared to Waterfall produce better quality?

32:47 PARTICIPANT 38:
I talked about that, before that Agile helps to write the software as it meets customer expectation, and this is the quality. But also, we didn't mention the documentation. So, when you look in Waterfall, all knowledge is passed in documentation, it will be a requirement specification, it will be some technical specifications. For testing, we have test cases. So, we have a lot of documentation and as human nature, when you pass some information via some media, like paper or electronic documents, you can miss some important things and that will lead to misunderstanding on another steps of Waterfall. In Agile, first of all, this difference and real differences might much close so much, it's less so the likeness of losing some of our information is much lower than in Waterfall. And in Agile, there is no need to have such big number of recommendation.

34:47 RESEARCHER:
I thought the opposite because I worked into Waterfall for years. I was a software developer before I come to academia, and I thought that I worked in Agile for a short period. So, I'm not very, in the industry practice wise, I don't have much experience with Agile. But I thought that losing documentation, you lose some level of knowledge. How do you compensate for that knowledge in Agile?

35:22 PARTICIPANT 38:
You have code. So right now, modern languages and modern tools, are very different from what we had twenty years ago. You can find the necessary information from the code much

faster and easier. So, you don't need to spend a lot of time on writing and with the training recommendation.

36:01 RESEARCHER:

But that's a technology feature. That's a technology value, but it has nothing to do with Agile, right?

36:18 PARTICIPANT 38:

So, Agile, I mean, that in Waterfall, you have very strict barrier between process steps. And their state might be this different steps, usually do different teams, and you need to pass information from one step to another. I mean, that in Agile, you have one team, and all knowledge is circulated within this team, and there is no need to pass this information. It's inside, it might be no formal documents, it might not be some website on Confluence, or wiki. And you store that information inside as a team, and there is not any information loss when you move further.

37:15 RESEARCHER:

So, it's peer to peer exchange of information, right?

37:19 PARTICIPANT 38:

Yeah. Absolutely.

37:22 RESEARCHER:

Okay, fantastic. I don't have more questions. Thank you very much. It was very insightful and very good examples. I liked your example very much. Do you have any questions for me before we finish the interview? If you want to know something about what I'm doing, or you want some clarification about?

37:45 PARTICIPANT 38:

Now, what will be your results?

37:50 RESEARCHER:

It's a research report. If you are interested, I can send you the research report. A copy of the research report.

37:57 PARTICIPANT 38:

Yes, please. That will be very interesting.

37:58 RESEARCHER:

Yeah, just give me a second, I'm going to write a note next to your name. So, I won't forget. Yes, PARTICIPANT 38. I'm also running a workshop with a few developers to validate the findings. So, what I'm going to do once we draw the initial conclusions, I will run a workshop with few developers. I will present the findings; I will get feedback from the developers whether the finding on the conclusion I made sense to them or not. If you want to be part of it, I can get in touch, and we can invite you to be part of it. Let me know and I will send you the invite.

39:03 PARTICIPANT 38:

Yes, please.

39:04 RESEARCHER:

Okay, fantastic. If you don't have further questions, thank you very much.

39:12 PARTICIPANT 38:

Yes, thank you.

39:12 RESEARCHER:

It was really very good. I like the example and it was really good. Thank you.

39:18 PARTICIPANT 38:

Thank you.

39:21 RESEARCHER:

Okay, I wish you a good day.

39:22 PARTICIPANT 38:

You too.

39:22 RESEARCHER:

Bye.