00:44 PARTICIPANT 34:

Hi RESEARCHER.

00:46 RESEARCHER:

Good morning, PARTICIPANT 34. How are you?

00:49 PARTICIPANT 34:

I'm fine, thank you.

00:52 RESEARCHER:

Just give me a minute.

01:02 RESEARCHER:

Okay, I'm all set. Thank you for the opportunity to talk to you and accepting the interview I really appreciate. I'd like to start with an introduction of myself and explain to you why I do the interviews, and etcetera. And we'll take it from there. My name is [redacted]. 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 mainly interested to learn how software teams achieve software quality; how do they work in order to achieve software quality. Currently, I'm running this project I'm trying to understand how as a Scrum, in particular, helps teams to improve or achieve software quality. We do interviews because we like to understand people experience people perspective. And we use people experience as a source of knowledge, and we capture it in our interview. And we analyze the interviews, and we draw some conclusions, and we report the conclusions in a paper or in a report. Basically, this is what I'm doing. And if you have any questions, let me know. Do you have a question for me before we start?

02:44 PARTICIPANT 34:

Not right now.

02:45 RESEARCHER:

Okay. So, the structure of the interview is I do have a set of questions, I will go through my questions. And it is fluid. Although I have questions, if you'd like to add anything at any stage, just let me know. Okay. How about if we start by an introduction, introducing yourself? And briefly a little bit about your software development process?

03:17 PARTICIPANT 34:

Yeah. So, my name is PARTICIPANT 34. And I'm from India. So, I live in [inaudible]. So, coming to the I.T. profession, like I am having, overall, like six and a half years, it's close to seven years, seven years of experience. So, I have been into both service based and product-based companies, like my first company was [redacted]. And the second one was [redacted]. And the third one is now it's [redacted]. So basically, what I do, like, we follow both Dev and testing, like, when I say testing, it's not complete testing. It's just unit testing. We do develop the scripts, and we do maintain them, and we do the regression. So, it's like, you can call it as a full stack one. So basically, we do both, and coming to the Scrum or Agile part, so it also depends on the project. And it also depends on the companies. So, when I say the companies, like let's, for example, take [redacted], where in some projects, I followed a pure Agile methodology. Like we used to have daily Scrum, and we used to have retrospections after the end of the sprint, so everything was strict. But for some projects, we don't have anything like that. It all It all depends on the Scrum master and the project requirements basically. So, if the project thinks that, okay, this is something that is really valuable to the customer, yes, they will follow all these sources because they don't want to mess up with any small issues. That's, that's why they keep this process. But I really like that process where they have this daily meetings and they have this retrospection, everything.

05:30 PARTICIPANT 34:

So, what the process usually happens is, like, let's say, today is the start of the sprint. So usually, the sprint will last for two weeks. That's what I have been, have been to. So, in the start of the sprint, so we have this grooming session where, like, every developer will be assigned to each story. So, we will be having some conversations like, what is the story about? And what are the acceptance criteria? And what are the story points and all. So, after that first day of grooming session, the following days will be like, we'll be doing development and, in the evening, or in the morning, it depends on the onsite team, like if the team is completely at the local time, so we will be having a stand up at the morning. But if they have any other regions, we will be having we can say that stand up will be having that in the evening times. So, what will happen there, like whatever the things that we have done today, we will be discussing over that. And whether the, it's like we have developed some, some piece of code, but we also need to test that. So, we'll be coming up with some questions like whether we have any issues or whatever the things we have to do. I mean, what are the things we had to do to fix that. So that's the process, like every day it happens. And we get feedback from our manager, team leads too like, what can be done well, so this process renews. And before the end of the sprint, we'll be having a session where we'll just record like everything went fine or not, like whatever the things that are not, that we cannot be delivered, we'll be moving into next sprint. So, we will be having that track of records. So, after the sprint, we will be having the retrospection meeting, where we will be discussing like what went well, and what can we improve? And how can we achieve this? So, these are the points. Here's how the process is.

08:10 RESEARCHER:
So what do you like about Scrum or Agile in general? What is good about this process from your experience?

08:21 PARTICIPANT 34
Yes, see, so, I can tell you this, taking an example, like, in the past, like we have this traditional development thing, see, like, where the product was developed completely, and after, like, two months or three months, we will be showing it to customer. And then if the customer likes it, yeah, it's well and good. But if he doesn't like that three months, time is totally wasted. But here at Agile, like we'll be doing in a strange way, like let's say I have developed in some small fraction of code for two weeks, and if the customer really wants to see what the progress is, he will see that and he will give the feedback. If they like that, if the like the idea or they like the development process, he will say, Okay, yeah, I'm good. But if it doesn't, like he will straight away say like, no, it's it should be improved. So here we will be just we can say that. The time we study is only just one string. Instead of wasting the whole three months or six months, we'll be just wasting one or two weeks. But we can get the

inputs and we can work on that later. So, this helps us both customer, both from customer perspective and developers time.

09:51 RESEARCHER:

So let me interrupt and have a follow up questions. This is a good quality that Scrum brings to the process which is frequent feedback. So, what do you tell what you told me is this frequent feedback ensure that the product meets the end user expectation? Right?

10:13 PARTICIPANT 34:

Yes, yeah.

10:13 RESEARCHER:

So, does it help the quality this frequent feedback from the client?

10:19 PARTICIPANT 34:
Yes, technically, because in what, what is the end goal, like we have to satisfy the customer, that's what. So, although there might other there is no bugs, or the purpose is completely fine. I mean, like, the purpose is excellent. But the customer might not be interested in the design, like, he will, he will come with one expectation. But if we give him a product, which doesn't meet his expectations, it's not good, right? So, the sprint, it helps in solving that. Because if not one sprint, we have other sprints like, we can develop our we can get inputs from the customer again, and we can work on that. And based on that, the quality team also does the same. Like we'll be as soon as we get the, we get the inputs from customer, we'll be working on that. And the same way once our code was moved to QA. So, they will be testing it in parallel and they will be providing the feedback. So, it helps us in achieving the product in quicker time and also helping to meet the customer demands.

11:50 RESEARCHER:
Okay. I will move to the next question. So, in the upcoming question, we will be talking about quality, we already started talking about software quality. From your experience and your perspective, how do you define software quality?

12:16 PARTICIPANT 34:
Well, as I said, like, not, not a single person writes the code, like for complete project. So, what we'll be doing is whoever writes the code, like right now, we are using Git to push our code so that at a time multiple people can work on that. And once the code was moved to the Git repository, we will be creating a cross review, where we'll be adding some SMEs and our team leads and other peers just to review the code so that we are not missing, like, we are not missing any standards. So, what I mean standard is there are some things let's, in a technical way, what I can say is, instead of not making the code structured way, like let's say, if I'm having a complete, like I belongs to Java background. So, I can say like, if my Java page is having too many debugging statements, like, sys out and system that print, and many, many things like that, so it doesn't look great, although you're debugging, your writing for that for that debugging purpose, you have to remove that when moving into the code review, because that's the thing that's going to move to production later. So, this way, like whether we have all these method names, and variable names, everything is in proper things. So, these things will be reviewed by other peers. And sometimes even other teams. Like we'll be adding one external reviewer too so that he can provide his own comments. So, this way we'll be having this cross review, which helps us achieve that software quality.

14:27 RESEARCHER:
So you've been talking about software quality is like if it was code quality. So do you, is it my understanding you see software quality as code quality only right.

14:46 PARTICIPANT 34:

That is a first priority for us?

14:49 RESEARCHER:

Yes, because you are a developer of course. Okay, fantastic. I move to the next question. Now we have defined what do you mean by quality. The next question, you already answered some of it, which is, so I will ask it and clarify a little bit. So, what do you do in a Scrum team to assure software quality? You talked about code review, you talked about following standards. What other techniques and tool do you use in your, in your experience or in your team in order to assure quality?

15:36 PARTICIPANT 34:

Well, tools, we generally, like, we have our integrated plugins, like we use different IDs. Like, again, it depends on the companies, like, in the past, I have used Eclipse, and now I'm using IntelliJ. So, we have sometimes we have our integrated plugins, like where, in our company itself, they have developed that. So, what I can say is code coverage. So, there is one plugin that we have, where it will take all the, what is it called, like initial steps, like whether the lines mean, whether the code is not too much. What I can say is.

16:31 RESEARCHER:

Are these linters of these linters you're talking about?

16:38 PARTICIPANT 34:

Which ones?

16:37 RESEARCHER:

Are these linters? Or they check the code? They are static analysis, right?

16:44 PARTICIPANT 34:

Yes.

16:47 RESEARCHER:

Okay.

16:47 PARTICIPANT 34:

Yeah. They take the format. And like, how many characters, and how much length of the pile. So, everything. These are the basic things that they check? And what was the question again, you asked?

16:59 RESEARCHER:

What quality assurance tools and technique you use to help with the quality?

17:08 PARTICIPANT 34:

Okay, another thing I forgot, like, we have this automation suites that runs on a regular basis, where it helps us to check whether the application is stable or not. So, we have these, we have this category as regression test, a smoke test and all. So, it will basically, does the basic functionality check. Like we both have UI and API automation, where the scripts will be running in Jenkins, so the major things are not working every time so like, I think it runs every day, on a scheduled basis. So, it helps us whether it's actually a bug or whether it's actually an intermittent issue. So that's one thing that helps us.

18:13 RESEARCHER:

Yeah, so who does the end user testing, is that the QA or the user themselves?

18:20 PARTICIPANT 34:

End user testing, I'm not sure about that. Because once we move to QA, that's the last thing that we do, like after the QA sign off that, then we will be moving into production.

18:38 RESEARCHER:

So, the QA does the final testing, right?

18:43 PARTICIPANT 34:

Yeah.

18:44 RESEARCHER:

Okay. That's great. I mean, you have quite advanced QA practices, which is good, good to hear. In your opinion, do you think Scrum help produce in better software quality? And how it does help?

19:03 PARTICIPANT 34:

Well, I can say, technically, technically, I mean, like, we don't follow the perfect Scrum, but still, we follow the set of things that actually has, like what we say, what I can say is, although if you are not having as a daily Scrum regularly, you still have to make your deliverables. It's you in the end who has to deliver your deliverables. So yes, Scrum helps a lot because there are some cases where in my previous organizations, I used to work with other company teams like I'm from [redacted] and my other colleagues were from [redacted]. That way the communication will be like, a bit tedious, because we cannot sit together, and we can work. So, what we have to do is we have to split the task and we have to, you have to work on that, and who does the thing, obviously, the Scrum Master has to has to decide, like, these are the things that you need to split. And, again, there will be a lead who will be telling us like, how you're going to split it and how you're going to communicate. And there are some even collaboration software's where we can actually like, zoom and like Skype, where we can communicate, and we can test, or we can develop live.

21:05 RESEARCHER:

So, you refer to this example, you not sitting in together, it didn't help much, right. So, sitting together, how it does help improve software quality or code quality?

21:24 PARTICIPANT 34:

Yeah, I can, I can say that. So, just take an example. Now, since everyone are working from home, so what we do is any, if I face any issue like this, I have seen another piece of code, where I want to use it in my project, in my module, I can say, so, I have to reach out to that person. So, what I have to do is right now, I only have an option to call him via zoom, or Skype, and to share my screen with him there to ask him to explain. Although there might be a little bit like, obviously, we will be getting the inputs from him, but still, there might be a delay in getting the communication properly, because it might be due to some internet issue or might be due to some, there are many other issues that we have, but if we are in person, like if the person is sitting on the same floor right next to me, I can straightaway go to him. And they can ask him like, what is this? And can you explain this, so he will be writing it, he will be training me in person. So, in person communication and virtual communication, I think there is a there is a far different way, which will help.

22:55 RESEARCHER:
So, I'm going to challenge you and pushing you to give me a more concrete example. So, working together facilitate communication, right? And you collaborate better. Do you have an example where this closeness or this better communication has helped you to improve your code or to improve the software quality?

23:23 PARTICIPANT 34:
Yes, like when I was there at [redacted]. So, I had to like, one time I had to reach another developer who is from another team. So, although he's not on the same floor, he is on a different floor. So just I just had to had to send a chat message on, I think we are using Skype. Yeah. on Skype. So, whether he is available or not. So once as soon as he told he's available, I can straight away Go to him, and I sit beside him. And I asked him like, what is this? And what are the change sets that you had? So, what does this do actually, and it's a piece of code where it actually, an integration part, like I have to integrate an e-commerce plugin to my application so that it communicates both way. So, although it took one hour time for him, for me and him to get into that issue, it solved us, and it helped us solve that. But if that is the thing that was done virtually, I think it will take a day or two because I'm not sure like what he's doing at home, like whether he's there at his workstation or whether he sick, I doesn't even know right. So.

25:08 RESEARCHER:

So, in this example, your colleague shared with you his knowledge and that helped to solve a code issue and subsequently improve its quality. Right. That's my understanding, right?

25:24 PARTICIPANT 34:

Yes.

25:24 RESEARCHER:

Okay. Fantastic. We will move to some rather challenging questions. You talked about some Scrum qualities, which is sprint planning, grooming, sprint review.

25:50 PARTICIPANT 34:

Retrospection.

25:51 RESEARCHER:

Sorry?

25:51 PARTICIPANT 34:

Retrospection.

25:53 RESEARCHER:

Yeah, the retrospective there. So, for example, how does sprint planning help you to write better code?

26:05 PARTICIPANT 34:
Well, see, we have a sprint for two weeks, each sprint for two weeks. So, what we do is like, based on the acceptance criteria, so we have one week, or like, it, again, depends on the story points, let's say if it is five story point, three for development process and two for QA process. So ideally, that five story point in the sense is like forty hours, that's what we can say. Like, it's each point, calculate says eight hours. So, what that means is, as I am a developer, I will take twenty-four hours to develop this product. And I will give you sixteen hours to test this product. So that you have to give me like whether in that, assume that the sixteen hours is for QA, and he doesn't consume the sixteen hours, and give me the like, give me the feedback in the end. Like, just one day before the sprint, he should not leave that way, like, Okay, this is an issue that I'm seeing. So, if we find an issue, he straightaway, this is a bug, so that I will be working on that bug. And he can move to the next task, which doesn't stop him from his activities. So as soon as he raises a bug, and I will be working on that fixing, fixing that bug. So, he will be moving onto next story, our next module where we can work on that. So, this way, both developer and QA, give me one second, I think my screen. Yeah. So, this way, both the developer and QA is not wasting time. And we can achieve that deliverable within the sprint, like, if the if the issue was not able to fit come in, like if the issue is major, and it will take some time to fix. So, what we'll be doing is we will be putting it into a backlog, or we'll be moving it into next sprint, so that we don't need to bother about that.

28:47 RESEARCHER:

So, what you're describing is the ability to properly estimate a story, make you more efficient, right?

29:02 PARTICIPANT 34:

Yes.

29:02 RESEARCHER:

Yes, but how does that help you write a better code or a quality software?

29:10 PARTICIPANT 34:

Based on the feedback that he gives either the QA or the other development team or the development peers based on the feedback like when I say like sprint, as I said, right, like, at the end of the sprint, we will be having these meetings like a sprint review, where the product manager, I mean product owner, will be the one who gives the feedback. So that we will be improving that based on the inputs that he provide. So, it's nothing but restructuring or rewriting the code like where we were lagging in. So that is the that is one way where we can improve the code, into the quality of the code that we write.

30:04 RESEARCHER:
So when you estimate your own work, does it give you a better opportunity to cater for time for to invest in quality?

30:16 PARTICIPANT 34:
Not exactly. Because assuming, like, obviously, since I'm developing the code, what I what I assume is, whatever the code I'm writing, yeah, it's good, because I tested it. And it's working fine. But it's, but it's not the case for others. Because I might be thinking it only in my perspective, like, I cannot have multiple combinations, like whether this code is actually good enough. Like, okay, I can tell you an example here, maybe just take a sorting example. We have multiple sorting techniques like verbal sort, Insertion Sort, merge sort, quicksort, and all. So as in that I have written a bubble sort code. So, I think like, Okay, this is the best sorting mechanism. So, we can straight away Go, and we can move it. But there's another developer who comes and say, your bubble sort, is good, but it's not efficient. Maybe you can use the mode sort, you can use heapsort. So that it, it will be better in terms of complexity. So based on other developer inputs, so we can improve our code so that the end goal is like, we have to produce a quality code.

31:58 RESEARCHER:
Yeah, I understand. So, sharing knowledge with your colleague helped you always to evolve your capacity to write a better software and better code. Yeah. So, you mentioned another quality, which you have in the Scrum process, which is grooming. I think it's a backlog grooming, right? That's what you're referring to, right?

32:21 PARTICIPANT 34:
Yeah. When I say grooming, what it does is like, let's say today's start of the sprint. So, everyone saying their stories, like I was assigned some two three stories, and the other person was assigned to other different stories. So basically, what lead, or the product owner does is, so he will be going to each and every story. And he will be explaining it like, What is this about? What are the acceptance criteria that we have to meet? And yeah, so like complete details. It's not liked a complete information. But it's, it's just information, that's enough for you to start working on that.

33:16 RESEARCHER:

So, this sharing of business requirements and understanding of the business requirements, you become more accustomed and intimately familiar with the stories, right?

33:33 PARTICIPANT 34:

Yes.

33:34 RESEARCHER:

So how does this quality of Scrum help you to write better code or to write better software?

33:43 PARTICIPANT 34:
Well, it helps us in saving the time on spending, what is this story about? Like, let's say if I, if, if there was no grooming happened at all. So, what happens is the developer will be in a confusion and the QA will be in a confusion. So, there is one example I can say like, okay, there was a sprint that started. Now, we don't have any grooming, but developer knows what he needs to do. Assume that developer knows what he needs to do. So, he straight away worked on that. And after that, he moved into QA, but QA doesn't know what that is. He need to ask the developer again. Can you explain me what is the story about, what I have to test because I don't know, and you straightaway gave me this? You straightaway move to QA, and I don't even know what I have to test on this. So, what you have implemented, what you have changed and what is the expectation? What is the expected response? Like, he doesn't know anything. Because if there is no grooming happened, everyone is like although the developers knows that QA might not know, and again, he comes to us and we will be asking to explain, which is not good at all, it will be a waste of time. Because, and also the quality, I think it will not meet the standards, because since the QA doesn't know what the story is about, and he knows only what the developer says, but what if the developer misses some piece of code, I mean, some piece of acceptance criteria, then again, the blame will be on QA.

36:06 RESEARCHER:

So, grooming facilitate more and better understanding of the requirements and the expectations. And this shared understanding between the developer, the product owner, the QA, helps fast and make better assuring that it meets quality standards, because the QA can find bugs, because he or she understand the expectation and this shared understanding of the expectation, you managed to pick up bugs and if the software doesn't meet the expectation of the user.

37:02 PARTICIPANT 34:

So that's what I told you.

37:10 RESEARCHER:

Yeah. So that was a good example. Thank you. Sprint review. Does it help and how?

Isn't sprint review the same thing as a retrospective or retrospective is broader. What's the difference in your team?

37:30 PARTICIPANT 34:
Yeah, I can explain. Sprint review is nothing, but we will be having a quick review. What I can say is, let's say today is the end of the sprint. So, what we'll be having is the day before, we will be having a small meeting from the team, like, this is what we need to achieve for this sprint. So, everything went fine, or does anyone has any issues, if you have any issues, so what we can do, whether it can be delivered by tomorrow, or we have to move it to backlog

or we have to move it to next sprint. So that's what the discussion happens. It's basically like, whatever the things that you have done a whole two weeks. So, we are going to just make sure that when I mean, the sprint meeting goes well. So, it's basically just one quick round of review as a review, peer to peer review and retrospection. On the other hand, is after the sprint, that was after the sprint ends, so we'll be having retrospection board, I mean, like, right now, we are using it online, but ideally, we will be having a whiteboard, we will, we will be writing it, like what went well, what didn't went well and how can we achieve this. So, these are the three categories that we have, like we have this, if it is went well, we have to add that like this is the one that was good. And this is the one that needs improvement. And this is the one I am suggesting. So, these are the things. Like it's nothing but called as acquisitions and improvements. And yeah, basically that's the one, retrospection.

39:47 RESEARCHER:

Yes. Okay, so my understanding is, a sprint review helps you to continuously improve right?

39:56 PARTICIPANT 34:

Not only improve, it's like what I can say is just to make sure that the sprint delivery, I mean that the sprint deliverables are on track.

40:19 RESEARCHER:

Yeah, so how does this ceremony of Scrum helps improving quality?

40:29 PARTICIPANT 34?
You mean the sprint review?

40:31 RESEARCHER:

Yeah.

40:32 PARTICIPANT 34:
What it does is it helps us, both developers and QA. What I can say is on the deliveries perspective, like, like, let's say, for the product owner. So, before the sprint, we are giving him like, these are the these are the things that will be delivered by this sprint. So, if it is not delivered, what we going to tell them is, although it takes a little bit time for us to move this product, I mean to get this delivery. So can you just give me some more time or can we move it to next point. So that's what it happens. Like, we'll be asking the product owner, some more time to give a quality product instead giving him a product, which is not completely, it is not complete. What I can say.

42:03 RESEARCHER:

What I've heard, it helps, because you, this communication or this closeness with the product owners, it's always gives you time to invest better on the deliverable and assure a better quality, that's what I understood, it's not a direct relation. It doesn't it, it doesn't make the code higher quality, but it gives you the opportunity to write better code and meet the expectation of the business. Right?

42:46 PARTICIPANT 34:
Yes.

42:48 RESEARCHER:
So talk to me a little bit about retrospective and how they are different from sprint review.

42:55 PARTICIPANT 34:
Yeah. So, let's say, today was the end of the sprint. So, the sprint deliverable was successful. So, we have delivered all stories, and everyone was happy. So, but we have to keep a track on this. Because, obviously, we have, we work on the sprint so hard. And we have a review, like, what happened, what went well? And what are the things that we have to improve? And how can we achieve this, so we have to keep a track of those things, as is our there's no point in just having a sprint and keep on moving on the same, same pace. So, what we do is we have this three categories, like let's say there are ten people in a team, including QA, there are ten people who worked on the sprint. So, after the sprint, so they will be given a retro board link, where they can write their own comments on this on these two sections. Like the first section will be like, what are the things that went well for this sprint? So, you have to say like, you can say, like, you can write like, Yeah, well the product met standards and the application went well. And the QA helped a lot and automation scripts help in identifying bugs on time. So, these are the things that went well. That's what we can write, and what are the things that we can improve? So, there's another category. So, what we have to do is like, if we see any, any things, I mean anything not like up to the mark. So, we can just write like this, this page can look better, let's say we are talking UI perspective, like UI design. So, what we can say is, instead of having this company page, the same colors, the same button colors, you can have one button a little bit larger than the other, based on the review. So, these are things that can be improved. So based on that we can write down on that category. And yeah, these are the things basically, what we will be doing is, like, it's nothing but like, individual comments, among ten people in the team, everyone has a right to comment, but nobody knows who has given that comment. So, this way, we are not making others feel like okay, this guy didn't like the way that I did. Because nobody knows, like, who has written that kind of comment. But still, they can improve on me on that thing. So, this way, it helps.

46:37 RESEARCHER:
Okay, it helps quality by identifying weaknesses and errors in the way developers work. Right? That's my understanding.

46:48 PARTICIPANT 34:

Yes.

46:47 RESEARCHER:

Okay, fantastic. We will move to some questions. I asked you for some examples. And this is one of the examples I want, how does a Scrum help you produce in high quality code?

47:08 PARTICIPANT 34:
Again, like, as I say, since we are as we are working on a team, so, again, it’s based on that review, code reviews, code reviews are the one that actually helped us in getting the quality of the code quality of the product. And also, there are some, there are some common suggestions that will be given by other developers, like once you write the code, and

obviously, if the product that you're working on looks good for you, it doesn't mean that everybody likes that. So, getting a peer discussion, is this application looks good for you? If not, can you just comment it out? Like, what are the things that happens? And also getting inputs from the lead or the manager, like, what can we improve on this page design? Or what can we improve on this UI perspective.

48:26 RESEARCHER:
So this could be done outside Scrum code review and could be done outside Scrum. What does Scrum bring better to this code review, for example? Does it make it a better practice? Or does it help or, and how?

48:48 PARTICIPANT 34:
Because that's what we do we even in Scrum, because when we, when we do have daily Scrum, that's what other people say is like, why can't we do this way? Why can't we like, ideally, in the Scrum, what we have is, whatever the things that I have done, we have to show that we have to showcase that and if it looks good, nobody talks. But if it looks something not right, then people might comment on that, like, why can't it be done like this. Usually, it will be done by higher people, like people who are at all levels above us. So basically, those are the people who actually does that.

49:47
So my understanding is the transparency that Scrum brings to the teams make you more accountable and the writing, that accountability to your teams drive you to write better code.

50:12 PARTICIPANT 34:

Technically it is. That's what it is. Because in the end, after the Scrum meeting was done, everybody will be like, again, there will be like a peer-to-peer discussion, like, you can do this way you can, you can just move this piece of code to here, and you can use this one so. So, basically, it is the sessions that other people use.

50:46 RESEARCHER:

Okay. Fantastic. I need more examples. Does Scrum makes finding bugs better? Does it help for example, finding bugs?

50:59 PARTICIPANT 34:
Yes. Again, when I say finding bugs, it's not only the Scrum that helps, but also many things. It's many things. Because as we developers do unit testing, so we think in our perspective, like, we don't have multiple checklists, like, these are the inputs that we had to give. And these are the outputs that should be expected. So, we'll be having limited things. But what the QA people does is they have multiple combinations. That's what their job is too. So, they will be identifying the bugs, and they will be raising the bugs in the JIRA where the Scrum master in the meeting will be bringing it under the point like, these are the bugs that I'm able to see. So, what's happening, so, and again, this automation, that also helps in identifying bugs. So again, like, if you're getting too many bugs, then you have to work on your product, like work seriously, because too many bugs is not too good. And too little bugs is also not too good, because there's no way that a product can have zero bugs and can go to the production, there is no way that any software without zero bugs is there in a production. So yes, Scrum helps that way, like discussing like, what is this bug about? And whether is this

bug really a bug or it's just an intermittent one and it will go off? So, we have these discussions.

53:06 RESEARCHER:
So it makes it makes discussing bugs, understanding bugs across the teams the same, right?

53:16 PARTICIPANT 34:

Yes.

53:18 RESEARCHER:

How it does help resolve in bugs. For example, you discussed with the Scrum masters and the QA and you as a developers the bug, does it help you resolve the bug?

53:35 PARTICIPANT 34:
No, we have to do we have to fix the bug. So, after the discussion, since we have this steps to reproduce, so we will be working on our environment. And we will be trying the same, with the same inputs that the QA gave. So, if we see that, yes, we even we are seeing the same thing on our development environment, then we will be reverting on that.

54:05 RESEARCHER:
Because the QA gives you the right step. And once he or she gives you the right step, you fix the problem.

54:18 PARTICIPANT 34:

Yes.

54:20 RESEARCHER:

So, it does help. Your example show it does help. It doesn't help directly, of course, but your understanding of the problem help you to fix it.

54:34 PARTICIPANT 34:
Yeah, but technically, Scrum doesn't need to be involved in that because what the process that we follow is, let's say there is a bug that that is raised by QA. So, what he will do is, we have this JIRA to track the bugs. So, he will assign directly to our work queue where either me or one of our team will get assigned, and they will straight away work on that. Although the Scrum master doesn't tell that you have to work on this bug, but it's your job that it was this task was assigned to you. I mean, this bug was assigned to you, and you had to fix that bug. So, the Scrum is just a part where we are not lagging behind. So that way it helps.

55:27 RESEARCHER:
Yeah, it helps you find the bug quicker. Yeah, well, that's why I understood from your example, it makes the bug quicker and you fix it faster. And it helps you deliver a better product. I'm not biased. I just read in your examples, it helps but your judgment, it doesn't. So, it doesn't match. So, but it's up to me how to interpret that it's not up to you. So, the last question is, how does Scrum or the environment that Scrum facilitates, motivate you to write better or to achieve better code quality?

56:12 PARTICIPANT 34:
Again, I want to talk about retrospection. That is the one that motivates us, because that is where you will get appraisals. I mean, like, that's where, you will get praised, that is where you will get feedback. And that is where you will see like, what are the things that we can improve or whether you're outstanding, so that is the one that motivates. Ideally, the Scrum Master, or, I mean, the Scrum master doesn't praise anyone, because it's your job, you have to what is that made the deliverables. They will just say, Good job, but it's not that. But in the retrospection that is where the manager or the leads will be talking about. So, what are the things that went well, so it's really good.

57:12 RESEARCHER:
So peer recognitions, other developer telling you it's a good work motivate you to write better code, right?

57:21 PARTICIPANT 34:

Yes.

57:21 RESEARCHER:

That's what I understood right. I don't have further questions. It was an interesting conversation. Thank you very much. I enjoyed it. Do you have any questions for me before we conclude?

57:36 PARTICIPANT 34:
Yes, RESEARCHER, you asked. I mean, you said like you're researching for this Scrum things, right?

57:45 RESEARCHER:
Yeah, I'm looking whether Scrum helps to achieve better software quality. I mean, I've been challenging you and pushing you to tell me whether it helps. Yeah. So, I'm looking whether it's makes a difference or not. You much younger. I mean, you've been for seven years, but I've been longer, I've worked in Waterfall. And at the time, I mean, years ago, we didn't have these practices like code review automation, which they help. These are modern software engineering practices. But we didn't have these methodology that's collaborate and facilitate. Like you said, yourself communication, and closeness and sharing knowledge. Like you said, you can call your teammates, and he or she can come and help you identify the problem or make the complexity easier. So, I'm looking whether these things help the developers and the environment to write better quality.

58:59 PARTICIPANT 34:
Okay, so, I mean, like, you have gathered this input. So, what's next?

59:04 RESEARCHER:
Next, we analyze these interviews, we have our methods to analyze them, I'm not going to take you through the process, because it's academic jargon and scientific stuff. And we draw conclusions. We make conclusions about the findings, and we report them in our report or in a paper and we publish it. If you are interested to read the paper. I'm happy to send it to you. It's up to you if you want. If you are interested.

59:43 PARTICIPANT 34:

Yeah, sure.

59:48 RESEARCHER:
Okay. All right. So, I'm also organizing a workshop with other developers to present the findings, the result and gather feedback from them. If you want to participate, I can invite you to participate. But it will take time, maybe in September or I can get in touch via email.

1:00:23 PARTICIPANT 34:
We can do that.

1:00:31 RESEARCHER:
In the workshop is, I will gather a lot of people, few people, six developers, and I will present to them the results of these interviews and what do I think, and I get feedback from them whether I'm right or wrong. So, I can check whether I got it right or not.

1:00:57 PARTICIPANT 34:
Okay, so you're working as a researcher or professor?

1:01:00 RESEARCHER:

Yes, as a researcher.

1:01:15 PARTICIPANT 34:
Okay. Sounds good.

1:01:18 RESEARCHER:
So should I get in touch once I decide on the workshops or not? It's up to you. You can think later on if you want to participate or not?

1:01:31 PARTICIPANT 34:
Sure. I think I can. I can. Just email me.

1:01:38 RESEARCHER:
Great.

1:01:57 PARTICIPANT 34:
Yeah, sure. Okay. Yeah.

1:02:08 RESEARCHER:
Okay. Thanks, PARTICIPANT 34. I wish you a good day. Thank you. Bye.

1:02:13 PARTICIPANT 34:

Thanks RESEARCHER. It's nice talking to you.

1:02:18 RESEARCHER:
Thank you. I enjoyed talking to you as well. Thank you very much. Thank you.