00:17 RESEARCHER:

Good morning, PARTICIPANT 30. How are you this morning?

00:24 RESEARCHER:

Good morning, PARTICIPANT 30. How are you this morning? I can't hear you. Hello. I can't hear you.

00:48 PARTICIPANT 30:

Hello?

00:48 RESEARCHER:

Yeah, now, I can hear you. How are you this morning?

00:52 PARTICIPANT 30:

Hi, RESEARCHER, how are you? I'm fine.

00:53 RESEARCHER:

I'm doing well.

00:56 PARTICIPANT 30:

Sorry about that. It's really cold in the UK right now.

01:02 RESEARCHER:

Is it?

01:03 PARTICIPANT 30:

It is, it is. We are having very strange weather the last three months, normally in Manchester and this time it's really warm.

01:14 RESEARCHER:

Yeah, it should be warm. That's what I thought.

01:17 PARTICIPANT 30:

It's fairly cold and the meteorology system, they expect it's going to worse. The country, really, different weather.

01:26 RESEARCHER:

Okay. It is it is warm a little bit here in Denmark. But warm in Denmark, it doesn't get over thirty. That's the highest we get.

01:43 PARTICIPANT 30:

When I was a young child, I had the opportunity to visit Norway. Around this time.

01:52 RESEARCHER:

Denmark, we are in Denmark, I'm in Denmark. Ah, you visited Norway. Okay.

01:58 PARTICIPANT 30:

So, I visited there. I was part of some conference for environment. So, it was raining, raining cats and dogs, and the coordinator for the conferences like we are sorry for inviting you during one of our hottest summers. This guy was playing with us.

02:20 RESEARCHER:

Okay, let's get on to the purpose of this meeting today. I'd like to thank you for accepting to do this interview. I really appreciate it. I will start by introducing myself and exactly why I'm doing this interview. And we'll take it from there. Yeah, my name is [redacted]. I'm a postdoc at the [redacted]. And I do research in software teams. I try to understand how software teams manage to achieve quality, software quality. So, the purpose of the interviews is to talk to people, software developers, and Scrum or Agile practitioners and to understand how they use the method, Scrum in your case, you're using Scrum, to enable or to enhance their ability to achieve software quality. So, we talk to a lot of people, and we analyze the interviews, and we draw some conclusions. And if you wish, sometimes we organize workshop and we present the findings to our participants. If you like I can invite you to our workshop where you can see. Yeah, I will put your name just to make sure I'll invite you to the workshop. So, the workshop may take place in August, but we'll stay in touch. So, do you have any question regarding what I do and my research before we start the interview?

04:29 PARTICIPANT 30:

I think the only question I have is when talk to different developers, are you looking for developers from all of the different platforms or focusing on corporates or startups or any type of organization?

04:43 RESEARCHER:

No, organization-wise, I'm looking for diversity. So, we can see the differences and we can see also the correlation between the practices across all types of organizations. So, in the past I had interviews with people from [redacted], I had people from large [redacted]. And yeah, so I'm talking to people from all over the world and from all types of organizations. Okay, if you don't have other questions we can start.

05:22 PARTICIPANT 30:

Right.

05:23 RESEARCHER:

Yeah. Fantastic. So, we will start with some presentation. So, can you present yourself and tell us what do you do and your role within the Scrum team? And, yeah, just a brief presentation about yourself?

05:39 PARTICIPANT 30:

No problem. So, I think I'll start by saying, my name is PARTICIPANT 30. I'm a software engineer from the UK, Manchester. And I am the co-founder of a startup called [redacted]. It's been operational since twenty-sixteen. So, I found that while I was still at university, it's been my main business ever since. But even as I've been running the company and going to school, I also did a lot of internships at various different startups. So, I have a lot of experience in startups that failed, because of the workforce, and maybe managerial organizations and a lot of ones that are very successful. And I've also worked for corporate ones, where it was a short one for almost about a quarter of a year. But I've worked for different teams in different capacities. In some of the organizations, I was a junior developer, I'm just there to fulfill small roles. And for the startup, not only a developer, but I'm also in charge of the technical aspect of the business. So, choosing which technologies to use, ensuring the timelines are met, making sure the code quality is exactly what we expect. So, it's been a learning process for us. We've been able to see which methodologies work, how to utilize Scrum, where it does and doesn't fit in. So, for me, that's how I'll introduce myself.

07:05 RESEARCHER:

Okay, fantastic. I'll move to the next questions. What do you think of Agile in general, and in particular Scrum from your own experience?

07:21 PARTICIPANT 30:

Okay. So, I think I mentioned we had mentioned it when we were emailing. But what's happened before, when I was still doing, still founding the company, still trying to figure out how to get things done. We just used to work whichever way. So, when what comes you just break it down into your deliverable and you say we'll finish this in a month, without really setting this will be started on Monday. We'll finish it on Friday, then we review it, etc. So, we just worked, worked and worked. And for a time, we thought it was the best way of doing things until I had the opportunity of working for a startup that used Scrum. In Scrum, they used to do these weekly sprints of our product and iterate it and discuss. So, what I noticed after going through that experience was, it makes for me personally, it makes the goals clearer. So that instead of worrying about the entire project, and having to keep everything in mind, I get focused on what I'm building, or what the team should be doing at this particular moment. And focusing on let's get this one thing down, this one particular thing that we're working on, let's do it very well. And let's do it quickly and making sure that we can review it. And because of the broken-down nature of Scrum, it's easier to do code reviews. Because someone did a small thing, when you tell them, this isn't good, because you didn't do this, didn't do that as opposed to going through a whole month of code. Sometimes you don't even review it very, with a very fine-tooth comb. You just said this works, it's fine. But when you're doing Scrum for smaller sprints, you get to be more fine-tuned with your detailed inspection. For me that's the experience. It makes it easier for the team to focus on tasks.

09:25 RESEARCHER:

I'll get back with some follow up questions on that. So, but I'll ask the next question. Because we will come back to the details later on. Can you describe to me your Scrum team and in particular, how it does work and who does what etcetera?

09:46 PARTICIPANT 30:

Alright, so I'll use for this example, I'll used our business startup. So, what happens is my partner and we are the partners and still part of the development process. So, what happens is once you get a project and you onboard the developer teams, we'll split it into two partners. One partner now will be in charge of the project and the other will be in charge of quality assurance. What that means is, one partner will be the one doing communication back and forth to the client. I want this so we'll work on this and this and this. So, my partner or I, depending on who is in charge will break down the project into the segments for each individual sprint. Once they do that the developers put the tasks on the Trello board, so we need you to do this, all the tasks that need to be done are divided. And those that have checklists, my teams are placed. And what the developer does is over the course of the sprint, which is most times one week. The developer will work on the items, then on Friday, depending on how much work there was, we'll have a meeting where the developer shows the code, we see what you have done, that now the developer who was not in charge of the part of the client, who's doing the quality assurance will now go through it step by step with the code at an individual level. So, it's supposed to achieve this and this and this, let's look at the code that does that. Let's look at the code that does that. This could be more efficient, so in large, the card, because it will be moved from a user to in progress. And comments will be added just underneath it, like improve this and this, improve the service, or this is fine. So, it moves up. So, depending on the issues that need to be fixed, when we meet again on Monday, or within the course of the week, the next Friday, we are reviewing the previous items that needed to be corrected. And any new items depending on which things were done. And that process goes over and over and over. And periodically giving the client updates to let them know this is where the product has reached. This is how far the system is looking. So, with Scrum I'm explaining this from a high level.

12:03 RESEARCHER:

Okay, so do you have a Scrum master? Do you have a product owner within the team? And what's the relationship like?

12:14 PARTICIPANT 30:

So, what we usually have is we have we call in our own language, we call it a project lead. So, what the project lead does, is he is in charge of making sure what the client asked for is what the client is getting. So that's sort of like the head of the entire project. So, if anything goes wrong, or the client has any problem, like if any problem with the system or any deliverable that they feel was not met, is the product lead who is supposed to answer to those questions and make sure that when they are questioning the team that's doing the problem, the programming or any other works, not just the problem, even the documentation, they are the ones who are in charge of it. So that's what we have. We have a product lead.

13:02 RESEARCHER:

Okay. So, is the project lead the liaison between the team and the business?

13:09 PARTICIPANT 30:

There's no business. When you say business, are you referring to the client?

13:13 RESEARCHER:

Yeah, the client. Your client. Because you deal with outside client? Yeah. Yeah. So, he or she does, okay. So, for a requirement clarification, he or she is responsible to clarify the requirement to the developers?

13:28 PARTICIPANT 30:

Yes.

13:30 RESEARCHER:

Okay. Okay.

13:34 PARTICIPANT 30:

Yes, they are in charge of making sure that the developers know exactly what the client requires. But to avoid their work, or because the product leads also in charge of the documentation, invoicing, customer communications, again, the work of also doing code reviews, you pass on to him. And we've noticed with Scrum, what usually works better is that we have smaller responsibilities that we can focus on. So not only will they let the team know the overall expectation of the client, they'll tell now the software lead versus the project lead, who's in charge of everything about the project, then there's a software lead, who is now in charge of code quality, and deliverables. So now what they will do is the project lead will explain a high level to the entire team, then on a separate meeting with their software lead and break down in very detailed instructions, which means that now, a software lead is always going to break it down and do code reviews for the software team. So, the reporting goes from client to product lead to software lead.

14:43 RESEARCHER:

Okay, so what format do you communicate the requirement to developers or to the teams?

14:50 PARTICIPANT 30:

So, when we're doing the initial, the high-level representations, the project says we have a client, they need this and this and this. We normally have, let's call it an informal discussion, like a meeting where it's a sit-down meeting where all the people in the team sit down, and the project leads talks to the team and says the client wants this, and this and this. So, we want to use this technology. We want to use this software infrastructure; we want to use this methodology of communication. That discussion between the developers if there's any question that they have, the team brings it up to discuss some of the suggestions. That's the

format in which it's brought up. So that's the first thing for the project lead to the development team.

15:44 RESEARCHER:

So, there is no documentation, sorry to interrupt, there is no documentation?

15:48 PARTICIPANT 30:

At that stage, no. The next stage has documentation.

15:53 RESEARCHER:

Okay.

15:53 PARTICIPANT 30:

Because the next stage is now where the software lead talks to the project lead. So now, there's more of a discussion. The client would like this, the software lead and says that we can use this technology, we can use this technology, these technologies, but now when we are discussing it with the team, because the client at this stage, has already given us a requirements document, because we require the clients to write down not just say, to write down every functional and non-functional requirement they need. So, at this stage, when we're giving it to the software team, we break down their tasks on a Trello board, depending on what the requirements are.

16:39 RESEARCHER:

Okay, so you're writing it in a sort of a user story.

16:44 PARTICIPANT 30:

We don't do user stories. What we do, we will break down the entire system as far as we can see, into cards, into a list of items, into epics. Then, once we do the epics will now from each and every epic, create a new set of cards which define what should happen, what should happen, what should happen. But during the informal conversation is when we're doing the user story, just talking, the client will do this, then when they log in, they'll do this, that discussion stage is where we go through our user story.

17:21 RESEARCHER:

So, can you take me through the journey of a user story or an epic from the start to the development and to the release? How does it go through the process in order for this epic or user story to be released?

17:41 PARTICIPANT 30:

Okay, so what happens is in our team, is that, first of all, because it's just the partners. So, one of us, the software lead or the project lead, we listen to the client. Because even though the contact point with the client or the business and us, is one of us will still be present for the meeting both of us so that we don't miss anything from the client. So, at that stage, we

are taking notes, but we also require the business owner to once we finish the discussion, they have to send us a written representation of their needs. So, we'll sit down, and we'll say, alright, so the way we understand the system will be this. Let's say it's a portal. A multiuser portal or let's say an e-commerce store, it's the easiest to understand. We'll break it down into epics, so we'll say, we imagine this to have one epic called the customer journey. A second epic called the front end and the back end, then we'll have the admin section back end. We'll create it on a Trello board together. Once we created the Trello board, so now let's focus on the customer journey. For example, the front end. We'll now start creating cards and assigning them to the epic. So, we'll say, what needs to happen first, we'll go through, we'll open similar, let's say it's an e-commerce store. We'll look at ecommerce stores that we've done research on and say, okay, there will need to be user login, create a card, assign it to the epic. Next, they will need to have all the products, create a card for the epic.

19:26 PARTICIPANT 30:

They will need to have a product listing, we'll create a card, assign it to the epic. If it has some divisions like sorting, down filtering, we'll create the sub points on the card. Then we'll do that process over and over for all the epics that we come up with. Then now it will be our list of new section that is filled with items. Then now going forward will be just choosing this week, we're going to work on these sprints. Now we move that whole section of cards, we'll move them to, whether it be new issues then now the developers will assign them. This needs to be done by this person, this has to be done by this person, do this. So once a developer starts working, they are supposed to take the product, the card they're working on, drag it into in-progress. Once it is in-progress, of course they work on it, there's not much you can say there. They're working on it. Once they finish, they should move it to Under Review. So now, once we come have the final meeting on Friday, those are the things now we checking. We said, user login can be login. User authentication there they list them like done, done, done. And so, on Friday, Friday morning, or Friday afternoon, depending on how much work there was, we'll go through the workflow, and say, alright, which developers, every developer who was working this week on this particular project needs a physical meeting, we'll go through this and this. We do more of Zoom calls because of the pandemic situation, then I will go through the work, because we know what the client is expecting because we had the meeting with them and that will go through all right.

21:06 PARTICIPANT 30:

So, if you're going to be present for this one meeting, but it's not really as important as the client meeting, so if the project lead is missing from this meeting, it's okay. The software lead is mostly the one doing the software, the current quality assurance. So, we'll go through an extra step to do this. Tick this done, yes, no. So that process goes on for every single card and every single sub item. If there's anything that needs to be done, the card is removed back from review put back into in-progress. So, in that stage, the developer knows this was not done, this is done, and if there any new comments. So now they work on that over the course of the coming week and then review them. Because the number of items is reduced, we'll still be finding new issues. But new issues depending on, like relative to, like if the thing that hasn't been finished is blocking what needs to be done, we'll assign new issues. But if there are things that couldn't be done parallel or concurrently, we'll add the new issues. And the cycle repeats, continues until when we have a demo with the client, which are usually every three weeks or so then the client sees, and we get a whole new set of...because clients sometimes come up with things in the middle of the week. And then we add those on to any new epics that have been assigned. And we start the process over again.

22:30 RESEARCHER:

Who does the testing? Do you have a dedicated tester?

22:35 PARTICIPANT 30:

We don't have a dedicated testing. What we do is a developer is supposed to test their code individually. And during the code review, that's what guides us in the review, in the quality assurance process, not just checking this work, like they are supposed to check the login. Okay, the user isn't allowed to log in before they see this page. What happens if we try to login without using that page?

23:03 RESEARCHER:

Okay, next question. Is Scrum setup that you have in place, in your opinion, is it good? Does it work for you?

23:16 PARTICIPANT 30:

To answer that question honestly, I believe that what we have now works. But I do believe that, because I've had the privilege of working for bigger companies, I feel there are a lot of things lacking. But because we are founding the business ourselves. We've never been employed anywhere in bigger teams; we are more figuring it out as we go. So, I feel like there are a lot of things that we could be doing better. One of the things you've actually touched it is we recently did, like a team course on, on testing, unit testing, feature testing and beta testing. Because that's one of the things we used to feel like this is really, we see people do it a lot professionally, but we never got the opportunity to properly implement it. Because of the stage we're in in the business. We don't do it yet. But we are moving into it. So, what happens is, especially in the business, I'm in charge of code quality. My work is to always be looking at ways to improve how the Scrum team works. But for what we have especially in the market, I think I need to point that out. The clients in the UK market aren't very understanding of the coding process, if that makes sense. I'll explain the statement. If you tell the client, this is taking longer because we're doing testing, the client is like, but this testing raises the price of the products. So, when you tell a customer, this will cost you a bit more and a bit longer, the first thing they start saying is, no, I'll go to someone else. You guys are taking too long and you're too expensive. So, such things make us pinpoint what exactly to go into.

25:12 RESEARCHER:

Yeah, to summarize, it's your implementation of Scrum is a work in progress, and it's working for you the way it is. Okay. Fantastic. I'll move to the next question then. How do you define software quality in the context of Agile software development or for you as an organization, what does it mean, software quality?

25:39 PARTICIPANT 30:

So, for us, software quality follows three main principles. It has to have high performance. So that if we are dealing with ten thousand Records, or even a hundred thousand, the users don't have to wait at their computer for ten minutes, just for something to work. So, what that does is, we might opt to choose a more, quote unquote, complex solution, as opposed to simply just running three, four loops and it works, as opposed to just immediately taking a bit a few more days to come up with a better way to iterate through things, just what works

better and the high load. That's the number one. Thing number two is scalability. We're coming from where we used to be before we first started implementing it now. Developers write code anyhow. No documentation, there's no code quality, no, let's say consistency, like people, one is using three spaces. One is using four spaces, others put a new line after every function, all those different personality traits that come with the developers. And what happened was, when you give one developer, the work that was done by another developer, it scales. It takes so much time for that new developer to understand what's being done. And something that is supposed to take us through our weekly sprint, takes about two weeks just because a new developer is trying to understand what was before. So, for us, code quality also means scalability. If we grow this team, or if we grow the product, it doesn't start crumbling.

27:19 PARTICIPANT 30:

And I think the final one which ties into the second one is maintainability. Coming just back to the argument of that we've done it before, this is not just all the three things you just pick from experience. Before, we started checking what needs to be...just all the work we have a month to do, let's meet in a month when the project works. If we ever close that project, or let's say the client can't pay for some reason, or we need to suspend it, and they come back three months later. Because before we used to do that, that time, it will take you to understand what you are writing, it's you almost have to start afresh. That is a project we started afresh, because we worked with an external service provider, and they did their own thing. They didn't do a bad job. But it was so different from how we were doing it. We couldn't make edits anymore. So, we just abandoned the whole project and started again. So, when I say code quality, for us is the thing that matters. It has to work well in the situation. It has to be able to move from different environments, with different developers without any issue, and it has to be maintainable. If you come back to this code a few months later, even a few weeks. It shouldn't be what are we looking at? What are the metrics?

28:39 RESEARCHER:

Fantastic. That's a very mature definition of quality. We will keep in mind that definition in the next questions because we will get in details the questions about software quality. So, any answer coming is within the context of that definition, which fit within the overall standards, definitions of software quality. So, we've been talking about the setup of Agile or in your case Scrum in your organization. What do you do to assure software quality in this Scrum set up? What engineering practices, what software quality assurance practices and tools you have in place to assist you meeting those quality criteria you've been talking about?

29:38 PARTICIPANT 30:

So, for us what we do, I think I'll point out first of all the tools that we use, not to the principles, yeah. My partner and I, we have a computer science background. And we've always been passionate about business and quality work. So, one of the things we do very keenly is we follow a principle, like O-O-P has always been object-oriented programming has always been our core principles. And one of the ways we use it is that we tend to take newer entry level programmers as opposed to programmers who have been in the field for a while. See, we noticed that when you get new programmers, it's easier to inculcate them into an already existing standard, because they've just left school, or they're looking for work. And they've just learned programming. If we tell them that, whenever you create a class object, you have to destroy it. Whenever you start a new, whenever you create us a class service, you have to create a single such thing, we forced them to follow the O-O-P principle that we've learned, and we practice. When you get an experienced programmer, they tend to

have their own understanding because they've been in the field for a while. So, they come with their own understanding. And they tend to start conflicting like no, I don't do this, I don't have a good idea where I worked, I did this. So that's for the principle that we use O-O-P as a principle and any other principle that we firmly believe we discuss us only the two of us.

31:19 PARTICIPANT 30:

But it's my responsibility to come up with some new software tools. I come and say, I'm thinking we should like test driven development. When you're doing unit and feature. I went and did some research on it. And I was like, now, I'm thinking this is where we need to go so that you can save the time for testing. So now I come, we sit down, and we discuss, and he's called Alvin. I said, Alvin, we need to talk about ABC, I'm thinking we should implement it like this. He asks me why should we implement it? Is it useful? We'll discuss, yes, because I think ABC, then now we implemented. So that's for the principles. For the tools, we use linters. So, like every product has, like, for example, front end tools, we have an ESLint and a Prettier file that is set by the software lead. So, when the software lead says, this is what this product can look like. So now, two spaces, arrows of the parentheses, everything that they feel is important, and we tend to use, instead of using our own predefined steps, we tend to use the global presets. Like personally, we prefer the Airbnb standard, like from JavaScript. So, when you declare an ESLint, you set up the file. Then you say everyone was working on this project, you have to, by default, the system substring, lots of errors, if you don't use the linting tools that are provided. And so that's one of the ways we use to keep quality. We use linters that are set for the whole project globally. So that even if you create a file in a file in a folder and a folder in a folder, the same rules apply. Actually, they've been the most efficient, especially when it comes to front end work. That's been a very efficient tool for us. Because before some people like two spaces in the HTML, some people like four. That linter makes sure that everything, nothing compiled is a mess, those problems are solved.

33:18 RESEARCHER:

So far, you talk about object-oriented principles and linters. You also talked about code review earlier in the interview, how does code review work for you and how it is implemented in Scrum?

33:38 PARTICIPANT 30:

Alright, so what we do is, we use VS code as our default editor, we did the research. That's what we forced to use. And we use the live share extension. So, what happens is, it's Friday afternoon, we are reviewing sprint for fourth of April twenty-twenty to the ninth of April. So, we'll enter a Google Meet or Zoom, any video conferencing that allows us to see both the output and the end of the code. So, we'll go alright, so now, the software lead opens the Trello on their end, and they open the view for the programmer’s work. So, we go Trello item number one, can the system give the user login. Now the developer shows first of all the interface, shows them logging in, shows them okay, click yes, yes. Does it work? Fine. I'll try throwing an error there. Try removing the email address. Yes, yes, fine. All right. Let's go to the code and see. So, the programmer switches the view or the meet. Then they go right. Look at this. I did this here. Have you sanitized the input? No. You need to sanitize it. Put a note on the Trello with sanitizing with not trimming. Okay. What happens if I do this, are you using a database transaction, so all that thing, so for every item that was supposed to be done this week, that's what happens. So, the Scrums, they tend to take long as I tried to put the whole Friday, because we reviewed each and every bit of code, which is why in the beginning, I said, we don't give very big chunks of code at once. We give things that we are

given a whole day to review this work, you will be able to review it. So that's what the Scrum process looks like for us.

35:33 RESEARCHER:

Okay. So far, you've been talking about principle that helped you for the scalability of the software design, like object-oriented design. And you also talked about code review, and you talk about linters. These principles and tool help you for the internal quality of the software, what do you do for the external quality of the software, for example, the functionality itself to make sure that what the client wanted has been achieved?

36:07 PARTICIPANT 30:

So, for that, one, is how we make sure that it gets exactly what the client wants is when the client is giving us the project. And the client says, I want an ecommerce store, or I want let's see whatever system to measure my agricultural outputs with hardware devices, because sometimes we do hardware devices. They have to give us, it's not an option, the client has to give us a written document outlining all the user scenarios. So that when we are building, we are going back to this to this particular item and checking. The client says they wanted to be able to send reports within ten seconds, does it do that. So that's the fastest way that we can make sure on the externals. The second part is we meet with the client regularly, every three weeks, depending on the client. Some want it longer, some want it shorter, we meet. We show them this what we have so far. Let's have a discussion to see, the client goes to the system. Now this is between the project lead and the software lead. The developers are not, they can, we tend to invite them so we can see the process. And sometimes after some questions that we may not have the best answers and that's usually when it's why you did this. And the client, in those meetings, we are taking notes, anything that they say, I'm not liking this, this isn't what I asked for, or did you also add this, or I was thinking of a scenario where I conduct our business this might happen. So, during those conversations, we are taking notes and recording them so that when we go now for the next set of Scrums, we're making sure that the client was satisfied with what was already done, and what wasn't, we are implementing it as we go along.

37:57 RESEARCHER:

So just to recap what you say, in these frequent demo sessions, give you enough feedback to implement the feedback and consequently achieve better quality?

38:13 PARTICIPANT 30:

Yes.

38:14 RESEARCHER:

Okay. Fantastic. I will move to the next questions. So far, you've been talking about some software engineering practices like testing, code review, linter tools, and some design principle like object oriented. These have existed for decades before Scrum itself. So, what does Scrum bring to the picture in your experience, from your team perspective to achieve or to help you produce software quality?

38:52 PARTICIPANT 30:

Scrum, for me, speaking, first of all, with an individual before, because I've worked for many different companies, and I've worked as a freelancer, not just as a company, as a company lead, as a freelancer. So, I've been on boarded with many other projects, some that use Scrum, some that didn't, some I just do work, we'll pay you when we done. What I find, the truth is, if you do Scrum well enough, if you really implement it, and are very careful, and anyone is dedicated to it, the need for a project manager diminishes. Because what happens is, you already know exactly what you're supposed to do. You know when you're supposed to get it done, you know how it's supposed to be done. And you know, exactly by Friday, and to have finished this, it needs to work like this, this and this, the client is waiting for it because for the client, it means this and this and this. So, what Scrum does, is it gives you, instead of you as a developer having to focus on every single thing. And the problem with having every single thing in mind is that the quality drops, with Scrum you just have to think of this week, I have to do this. So, you can focus your attention. And when you focus your attention on that smaller subset of things you can do, the quality always goes up. That's something I can do. I can almost...it always goes up. Whenever you give the developers smaller bits of frequent work, the quality goes up, because the developer can focus on today, this table has to work. I will make this table work very well, as opposed to giving them a plethora of thirty things. In their thinking of because even when they're doing one thing, item one or thirty, they can't even take time to focus on it. Because they're thinking, I have to do to thirty-three, or thirty-four, or thirty-five, or forty. It makes the quality go up. And it gives the programmer less to juggle.

40:52 RESEARCHER:

Yeah, but this can be done without Scrum, right? Why do you need Scrum? What did Scrum bring? What's the add value that Scrum brought on to this? You could always segment your work into chunks and focus into small chunks without Scrum. I know this question is a little bit challenging. But I'm looking here for what Scrum brings to the teams in order to help them produce quality. I understand that user stories are segmented into small chunks. But this can be done outside Scrum.

41:38 PARTICIPANT 30:

Okay, I get your perspective. And there is some truth to that. In fact, there is a lot of truth in any organized person can divide their work and do it piece by piece by piece. But the problem is, this is just my opinion, you increase the room for error, when you do that. If you leave organization to the discretion of a large group of people, without any sort of external parameterization, we just tell them organized, especially for working in a team. If it's a one person project, or even a two person, you can organize it. But if it's a very large team just to organize the work the way it's supposed to be organized. Everyone will organize it the way they think is best. And you'll have very different results, something that you expected to be done today or some developer will think I didn't think that is important. I'll do it tomorrow for me, this is always important. The client themselves because only one person is talking to the client only, they know what's important. That's one. Two, is that the review period at the end of Scrum, because that's one of them. Something that is very important in Scrum, the testing part. If you just let people do whatever they want and there's no review, you will get a finished good, yes. But because of the lack of standards enforced, it might not be the best quality. Or at the very least, it won't be a quality that is consistent among the entire team because people are different that will have their own preferences. If you don't have a review where a single source of truth like a software lead, or like a design expert or like database engine, you don't have a single point of truth. Everyone starts creating their own thing. So, it's organized, yes, but not really because even organize it according to their own way.

Scrum streamlines that process and says you do this because this is important, or this is important that everyone focuses their attention based on some things. Based on a highest standard that's being set. And once they do their work to the highest standard that checks it again. For me, that's the differences Scrum brings to quality.

43:52 RESEARCHER:

Okay, I've heard few things I just want to check with you and feel free to elaborate more. So, Scrum's concept of small team, self-organized teams because I've sensed that you mentioned that the team to some extent organize itself. And the reviews at the end of the sprint helps you to some extent to deliver better quality.

44:20 PARTICIPANT 30:

Yes.

44:21 RESEARCHER:

Okay. I'd like to go back to something you mentioned earlier, which I'd like you to elaborate more. The project lead organizes requirement discussion with the teams at in order to understand the requirements. You mentioned that earlier. How does this help understanding the requirement by the software engineers and subsequently this shared understanding of the requirement, how does it help the software engineer to write better code or delivers better quality of the software?

45:10 PARTICIPANT 30:

So, what that does is, when you have that first meeting, and it's, in our experience, if only it's successful, well, when that first meeting is a bit more informal, as opposed to having a very formal suit and tie meeting is fast forward, it gives the developer a whole view of the system. Because this fellow will most likely be working on many paths throughout the direction of throughout the duration of the project. So, if they don't, if they have an awareness, what leaving in the works, even as they built, let's say, a service or they implement a database call, they're thinking, well, I better do this in a way, knowing that probably next week, I'll need to do this, or the client wants this. So, as I build this particular feature, as I test the login interface, I have to allow for multiuser login, because I remember the user mentioned in the first meeting, this is what the client wants. So that's overall understanding of what the system does, it gives the developer a sort of awareness of the long-term direction. When developers work, if you don't involve them in that process, which we've done before, and then you just give them work. This week, do this this week, this week do this. They do good code because you're reviewing it. But we noticed sometimes you have to go back on code. Because you're like, okay, this thing that you built, no, it is good. You haven't, you need to remove this section, because the next part of the system will need this.

47:02 PARTICIPANT 30:

So, they're like, oh, I didn't know that I didn't because I locked it. I didn't consider like, the hard coded apart, or like you didn't make it as dynamic as possible, because they didn't know how this is actually something that I need to implement. Having that discussion, they keep it in mind. They don't have to really develop about it. But there's a way of framing everything in mind, makes them more feature minded. This helps the scalability of the system and future requirements. And there's the second part is because its informal, developers feel easy. So, they can suggest, they feel they can suggest and go, well, so I

read an article about design principles. I'm thinking we do this service, or could we try implementing, I saw this recently that they released the newest version of PHP, could we change this system to PHP eight instead of seven point that makes the developer, it doesn't have a direct output on the work, but it makes him invested. And developers that feel invested in the process, even when you're reviewing their work, they aren't antagonized. They don't feel like just working, they feel some sense of ownership. And that sense of ownership and participation and value is what a developer, is the difference between a developer who when it gets to five pm end of the workday, they put off the emails, don't look for me. I'm done. A developer who feels invested, she tells them, we need to do this soon. do you mind giving me an extra four hours. That fast meeting helps them feel part of the project and they're more willing to give that extra effort. It makes them more keen on ensuring quality because they don't feel like it's does some work I have to do. They're more, they feel let me do a good job.

48:54 RESEARCHER:

Okay, fantastic. That's a great example. I will go through a little bit of details because we have thirteen minutes left. So how does for example, Scrum, the way you implement it help you find in bugs, for example. If you can illustrate that with examples would be great.

49:16 PARTICIPANT 30:

So, when you say examples giving a screencast? Or do you mean like they give examples of times it happened?

49:23 RESEARCHER:

Yeah, time when it happens and how Scrum enabled you to fix bugs, for example.

49:33 PARTICIPANT 30:

Alright. There is a project we've been working on. It's a school management system. So, there's a database and it's being consumed by a single page application. So, what was happening is the two teams are building it concurrently. Back end was building the management of SMS because it had SMS integration. It had remote user registration and it had a lot of things going on and the front end was just basically consuming based on mockups that have been designed and building. So, when I think the front-end designer wasn't, they hadn't thought, give me a few seconds to remember. They hadn't considered a situation where a school doesn't have students, so it doesn't have guardians, because when they search for students, it means the Guardian when you select them to search out, I don't know why the developer hadn't considered what if a school is new, they don't have finance. How would this incident throw an error? It should throw an error, but it shouldn't throw those blocking errors like a 404 page like this, it should just say like, this cannot work because of so and so. So, when we are having the Scrum meeting, like now, they shown, they shared their screen, they've showed the design, sorry, the front end, and they're showing the code. So, you just let the developer demo at their pace.

51:11 PARTICIPANT 30:

So, they say okay, they're watching. Then my partner because we in the meeting together. At that time, he was the software lead. I was a product lead. He goes, just remove the clients we see. Just delete all the guardians, we want to see how that works. Then the developer deleted the guardians. When they deleted, the system crashed. The form that was not okay,

not the whole system, the form that was collecting that data it crashed. So, the developers like, oh, I haven't considered a situation where, yeah, so you have to consider that because the developer is thinking from a developer perspective, not really business. So, they were like, yeah, you have to consider some of the schools we'll go to won't have parents with registered phone, so now, go back and make sure that this code throws an error that looks like this when the school doesn't have any guardians. So, if we didn't do that meeting, you just let the developer, if you let them check themselves that yes, this worked, yes. Because from the perspective, it can send, it can send notifications if you just let them check themselves without that quality review. Without that, wait, let's test this thing, make sure it works completely, then roll it out, that would have gone to production, that was there. So, for us Scrum, just such things happen during Scrum, when you're having the Scrum meeting, reviewing the work, such things happen, because the different perspectives and developments have occurred in those meetings.

52:45 RESEARCHER:

Okay, I'm going to throw at you another challenging questions. I know the answer you already have it in the story you shared with us, and thank you for that, but I will throw at you a challenge. What new here about Scrum, I mean, a team that can collaborate and have close relationship under the same roof can achieve that. Can what you shared with me is a story where the developer, a lead, share his requirement knowledge and his knowledge of the business with the developer, and guided him to fix the error or the understanding of the requirement. So, what's new here? I mean, we could have done this without Scrum. I mean, yeah, so I need you to point out exactly what's the value add of Scrum in here?

53:46 PARTICIPANT 30:

Well, you see now for me again, first of all, I thought this is a very valid point. Because what you said is true. And I think there's no, there's no perfect system. So, I think if I was to analyze just the value being my own objective truth, or Scrum is the benefit of Scrum lies in the way it compartmentalizes focus. Yes, you can have a team sharing one roof, they have frequent meals together, they thinking of doing this, they go that, but I find that Scrum in my experience and how it's helped us as a company is you don't have to, I know that's a bit controversial. Yeah, it's controversial because you're telling the developers to not focus on building a wholesome system. But in a way, they're like, you don't really think of every single instance, if an instance comes to mind because you want the workplace to be collaborative, you're able to ask a question, but don't really focus on everything at once. Build your system, if you don't catch something it is okay. Built to the best of your quality in this particular thing. So that when we have the meeting on Friday, and you're able to focus and give it your all, we can now have this situation of all I'm thinking, this can do this, we should have done this, this is a better experience. What Scrum does is, for me now I'll go back to this one point, it isolates the focus point for a developer. Think about this and implement this. Anything else that arises, is handled, is mitigated during the review. As opposed to having a developer think about this and this and this. I also need to think about this and this and this. No. We have a session for working and focusing on technical expertise, because the coding period is a period of technical expertise. Then when we come back to the Scrum meeting, when we are reviewing the code, in that season is when now we can now talk, I'm also thinking you can do this, I'm also thinking you can do this. Focus on your work, do the best you can with what you are aware of. If you feel something, ask, but don't worry, during the code meeting, we will grill you and we will grill each other enough to handle anything. So, that's what I think the biggest benefit of Scrum is.

56:18 RESEARCHER:

Fantastic. That's a great answer. We coming to an end, but I still have two questions. If you can answer them quickly would be nice. I don't want to take more of your time. So, the next question how for example, Scrum helps produce in high quality code. You are a software engineer, you write code every day, and your team writes code every day. So how does Scrum helps producing high quality code?

56:45 PARTICIPANT 30:

Well, the less code I have to write, the more I can focus on doing it better. That's my short answer. The less code I write, I focus on how I might make my code better. I always check its quality. The less code, the more I can focus on quality.

57:01 RESEARCHER:

Fantastic. That's a very nice answer. And so, the next question, how does Scrum environment motivate you to achieve high code quality as a software developer or a software engineer?

57:17 PARTICIPANT 30:

I'll give two answers. One, it's very subjective, I'm showcasing my code and it works very well on Friday. Very nice to have. His code is nice, don't worry. Like that feeling of yeah, because it's a public forum. So, when you're going to the code and there's no problem and everything is fine. It's good. You feel good. And there's no problem. And two, the benefit of Scrum, this is when I was working for other teams, and I was already a junior developer, is when things don't work as well, I don't feel like oh, no, it was my fault. Everything went bad because I made a mistake. It's like, because we're all in this environment of review. That review doesn't mean why did you get this wrong? It means I can see you didn't think about this. Next time let's do this. I don't feel attacked for being wrong. If that makes any sense. I feel less inclined to feel attacked. So, I'm more productive, because I can just focus on doing what I can to be more productive and better quality.

58:26 RESEARCHER:

Yeah. So, there are two things you mentioned, you become more productive. And the other things you mentioned is peer recognition. At the end of the review, they tell you that your code is good. So, in order to write good code, you need to be motivated to write good code. Is it achieving that peer recognition helps you to write good code recognition?

58:54 PARTICIPANT 30:

Peer recognition helps write good code. For two reasons. One, you most likely look up to them. Because you're all peers, you're all tried fix it, when they look up to you. And when they think you did well, you feel as I'm thinking in the second thing is you gain confidence. Because you're like, oh, so turns out, I actually can write good code. So that confidence of writing good code makes you write better code. So, when you display your code, and it works in front of your peers, you're slowly feel more competent. And that confidence produces competence.

59:31 RESEARCHER:

Is it having to do with transparency, because you can accept feedback and you can accept criticism? Does Scrum facilitate that?

59:43 PARTICIPANT 30:

Depends on the organization. I'm happy enough that the organizations that I deal with are my organizations we I've worked, the Scrum is a bit informal. It's not suit and tie. If you get it wrong, you feel like you've lost a job. It's more of Okay, so what did you do? So, if an environment is welcoming and understanding that people make mistakes sometimes then you feel confident going to Scrum.

1:00:10 RESEARCHER:

Fantastic. So, the last question is a little bit controversial, I may take two minutes out of your work. So, what would be your reaction to the statement Scrum or Agile produces poor software quality?

1:00:29 PARTICIPANT 30:

I think that speaks less to Scrum, in my opinion. I think it's less to Scrum and more to the organization around Scrum. Scrum is people oriented, it's deliverables and people oriented. If you give people tasks that are too big, or like tasks that are not catered to the competencies, is going to go wrong. Of course, because Scrum in a way, it makes it take a bit longer, because you're doing small sections. So, if you give someone thirty tasks, when ideally, they should be doing five, or maybe they should be doing twenty, it's going to slow down, and you're going to think Scrum is a problem. No, you just work with this fellow. And point two is that if, if the review is something to dread, if the code review is something to dread, this program, I want to be productive, because even as they're doing their work, they don't feel that programming is a very, for me, in most [inaudible] most of the developers I've interacted with, it's something that is you enter a flow, where things start working. I see developers working late into the night and they're not tired. But if you're anxious, and feeling this, will I get this wrong, I'm dead. And then that you don't achieve that state of flow. So, when you start handling your tasks, you don't achieve them. Or you just don't do them well. And now, of course, because you're dreading the Scrum, the review stage, it doesn't go and of course, more shouting, more dissonance. So, if this is for code, in my opinion, it's because it probably isn't being organized well enough.

1:02:13 RESEARCHER:

So, this...

1:02:15 PARTICIPANT 30:

Just to add something you can't, you can't copy paste it across all levels, like the Scrum, how Scrum is implemented at a startup cannot be how it's done in our corporate company. You have to understand the dynamics of the area before you start implementing it.

1:02:33 RESEARCHER:

So, it has to do with the implementation of the method itself. It doesn't have to do with the philosophy behind it, right?

1:02:42 PARTICIPANT 30:

Yeah.

1:02:44 RESEARCHER:

Yeah. Okay, PARTICIPANT 30, we come to an end of this interview. On time, actually, thank you very much. I'd like to thank you a lot. And I'll invite you to the workshop. And I will also send you the research report at the end. So hopefully, the workshop to discuss the results will take place sometime in August, but I will get in touch. Thanks a lot, PARTICIPANT 30. It was an interesting discussion. Thank you very much. Do you have any questions for me?

1:03:25 PARTICIPANT 30:

Yes, I have one question for you. So now this is I think back to you. Do you feel like this conversation you've had, I'm sure you had a lot of conversations?

1:03:34 RESEARCHER:

Yeah, I did. Yeah.

1:03:36 PARTICIPANT 30:

Much more competent fellows. And maybe people at my level.

1:03:39 RESEARCHER:

I did. I even interviewed CTOs and senior manager to see their perspective as well. Yeah.

1:03:49 PARTICIPANT 30:

Exactly. So, I just want to know, my opinions especially as a young audience, less experienced was that of any help to you like was their contributed anything else that has helped improve or change your opinion or just more of the same?

1:04:05 RESEARCHER:

There is a significant enthusiasm around Agile and Scrum and how it helps. It brings qualities to the people level. Like we discuss here, each time as I push you to tell me what Scrums brings into the picture, you always go to the people quality and the social qualities that Scrum brings into the picture or Agile for that matter. So that's the sense I have. As you know, the software engineering practices and the quality assurance stuff you've been talking about, that's not new. That's something have been advocated for decades in software engineering and software development process. But Scrum what people tell me what Scrum or Agile in general brought to them is the process and the people qualities. You talked about a shared understanding of the requirements and facilitating that shared understanding of the requirement, but by close collaboration across the teams that also empowers the software developers. And this empowerment, like you said yourself, you use different words, enables the software developer to write better code and to achieve better qualities. So overall, people are in line with you. That process-wise and people-wise, that's the value add of Scrum. And we can discuss that in detail in the workshop. So, I will be presenting this result in the

workshop, and you will have an opportunity again, to tell us what you think. But that it will be a different format, I will be presenting the result. And you and other software engineers can give me feedback. And what do they think?

1:06:18 PARTICIPANT 30:

I'm looking forward to it.

1:06:19 RESEARCHER:

Okay, fantastic. Thank you, PARTICIPANT 30. and have a good day. Of course, if you have any question, you can send me an email okay.

1:06:29 PARTICIPANT 30:

Alright.

1:06:30 RESEARCHER:

Thank you. Bye.