

Researcher

SUMMARY KEYWORDS

team, quality, mistakes, software, people, developer, work, fantastic, problem, tester, person, issue, agree, scrum, helps, discuss, bringing, retro, team members, system

SPEAKERS

Researcher, Participant 2

Researcher 00:14

Good morning. Hello. How are you this morning?

Participant 2 00:19

Yes, I'm fine. Thank you. And you. How are you doing?

Researcher 00:23

I'm doing very well. Thank you. I think we should start. I'd like to start by thanking you for doing the interview. I appreciate specially this Saturday morning.

Participant 2 00:34

Yeah, no problem.

Researcher 00:36

Alright, I'll introduce myself. And we'll kick off with the interview. All right. Yeah, sure. Okay. My name is [deleted to preserve the researcher/participant anonymity]. And I do research in software development teams. So I focus on how software development teams achieve quality and in the context of how do they work together and their social environment and how it helps them to achieve better quality. So I'm doing this study on the topic of feeling safe in a work environment, and how does it help the team and the software developers to achieve better quality? So we do interviews, and we do later on we planning to do a survey, and hopefully we can come up with some proposition. And we publish our work, as you know, because you worked for a research agency, I think so you might understand. Yeah. So that's it. Do you have any questions for me?

Participant 2 01:54

No, I just want to say that I think you are. You're actually I say, lucky that you chose me, at least for one of your subjects. Because, yeah, I start to introduce myself as well. I work as a front end developer. But also, I'm the same time a scrum master at a company, which the work is outsourced to us. So I'm working for a Dutch client, but we are a dedicated, outsourced team. And yeah, we had a lot of issues lately, and mostly because the quality wasn't very good. And also, we lack a couple of people, at least in between the managers and developers, maybe I can help you with a couple of answers.

Researcher 02:52

Fantastic. Because I selected and I approached you because of your answers. Actually, I liked your answers very well. Okay.

Participant 2 03:03

One question maybe if I can have sorry to interrupt you. For when I'm adding examples, should I make it specific? Should I say names? Because, you know, for the company, maybe I cannot say all the all the information?

Researcher 03:26

No, you don't have you don't have to give me names. But just give me an example that makes things clearer. But you don't have to give me names. Exactly. You don't have to give me for example, the software name the person name? No. But if I need more information, I will ask, and you don't have to give me the details and the names etc. No, you don't. So can you start with a brief and introduction? Can you introduce your education, your experience? And yeah, yes.

Participant 2 04:04

Yes, sure. Well, I'm a developer in the UK. So that's why my, my profile is said from UK, so I graduated in 2006, the university of Warwick. I'm not sure if you're aware of it. I was a with a profile of computer science. And in 2007, I started working as a developer. Since then I worked for few companies. And for this one, like almost one and a half or something like that. So I have like 14 plus years of experience. I've worked in many, many languages, even from them backend. I like most the front and now, which I'm working in at least, because I mostly see what I'm doing. So that's pretty much it. I'm not sure.

Researcher 05:36

That's, that's more than enough. Thank you. And I had a look at your LinkedIn profile. So I'm very happy. So thank you very much. So before we go into the details and go through your answers of the questions you provided in the emails, what type of agile methods do you use in a team

Participant 2 05:59

At the moment, we're using Scrum?

Researcher 06:02

Okay, fantastic. So can you briefly take me through the process? How do you use Scrum?

Participant 2 06:11

Well, yes, we started like in September, because then we had a major change at the company I work for. So the managers and the bosses so said, they were asked to leave because they didn't do a very good job. And since then, new persons came in. And before that we were working as not a non Scrum. But another process. And since then, we came back to Scrum. And we divided into three teams, I will mostly talk about my team, we have a team of five persons, two front end developers, one that can develop backend one UX and one tester. And I'm responsible, at least for the scrum mastering the team. Which was very difficult for me, because I had no idea how to do that in the first place. I was just

picked and asked if I want to do it. And I said, Well, yeah, sure, why not. But without any knowledge, we started to do so at first we had I will explain later because I saw their question as well that I will come back later. And since then, we are working with Sprint's of three weeks. And at every at the end of every sprint, we have a sprint review, which we are trying to prepare with a presentation to the stakeholders, we are presenting what we did in the previous sprint with explaining but also making a visual representation of making an example. So we have a definition of Done and then PBI because we PPS, the things we do when it's done. So based on the definition of done is done that means we are merged to our DEV environment. And then we can we can show a presentation and from that, other than that, we have the regular scrum things like blending, we have learnings for two times per week, one for a two hour session and one short with a one hour session. But of course we discuss before if we have anything to, to refine or not. Because there are some times that there are no more things to talk about, because we already did in the previous sessions, but that happened like one or two times. Not more. And other than that we have at the start of the next sprint we always have a read through about our previous sprint where we discuss our what went well what went wrong, the regular stuff. Usually I try to make it fun because no one likes the retros so I'm trying to bring in a couple of fun things there. And yeah, you should know that my team is like one person is really talk and the rest or like not really talking about it anything basically, they are really a good team. But when it comes to talking, it's a bit more difficult. I have to have to be rough and, and take the words out of people.

Researcher 10:16

Yeah, that's, that's we will get back to that. That's a problem, especially in retrospectives. Okay, let's get over the introduction bit and get into the details. So what type of software being developed?

Participant 2 10:31

We are doing a ecommerce projects. So basically, okay, as a front end and the backend.

Researcher 10:43

Okay. Yeah. We will be talking about software quality and the rest of the interview. And it is important to align our understanding of what we mean by it. We use ISO standards as a definition, I will read to you the definition and we can discuss it whether you agree or you disagree, and whether we can comment on this definition. So, the ISO definition of software quality says software quality is the degree to which the system satisfies the stated and implied needs of its various stakeholder and thus provide values. So the ISO model always cover some also covers some non functional characteristic of a software. So it has performance compatibility, usability, reliability, security, maintainability. And portability. First, do you agree with this definition? And would you like to comment on it?

Participant 2 11:54

Yes, I agree. And I also heard that you said that it brings value. And that's also important for us. And it also has like to waste of value, at least what I see is one value to the to the end users. Because you know, that software needs to be in a way that they would really like to be, and they can use it, for example, our project is used for, for a work for another company, so they work in it to, to actually do their work. It's a kind of software when you can, and you can mobilise your, your actual company. So it's like a model based software, but online, of course. And another point is value for also for us, because we should sell the product so that that's also value, at least how I see.

Researcher 13:09

Okay, fantastic. It is rare to see someone focusing on adding value, but it has become within agile team, they emphasize a lot adding value. And the whole idea behind Agile is adding value. I agree. Exactly. Yeah. So that's quality. I'm glad we agree. So what do you have in your team as quality assurance practices and processes to achieve this quantity?

Participant 2 13:42

You mean about the quality of the product? So it's actually working? Oh, you said all the things or Yes. For example, value?

Researcher 13:51

Yeah. Know about the quality and also the value? So for example, yeah. Yeah.

Participant 2 13:59

Well, this is a topic. We can talk for handful more hours.

Researcher 14:04

Yeah. Just give me some examples that yeah, that's it. That's all I want is what I and what I want to know is whether we align our understanding of what quality assurance processes you have.

Participant 2 14:18

Yeah. Well, we have issues also on the usability part. So the UX sirs, it was a small thing, and we just did something to be there. And we never, for example, we never checked with our clients if they really have a good feedback on what we did. So it's really usable. And I didn't want I don't even want to talk about if it's working or not, because we have also problems there. And we have issues also on the on the testing team. So the actual cause which means there are no bugs and everything is working. So the actual How should I say the product is working as described in the requirements. We also have problems there. Because of multiple reasons the team is small. Also, they are not, not senior people, only one of them. So, there are many, many issues. Also, we had before going to scrum we had an issue of everyone worked as hard as they could, it was like a headless chicken for running all over, everyone did what they picked up. And that's all and there was no person to, to have an overview what was done and, and how it was done. So before that this was an issue. Now since we're working in Scrum, there's more control over everything. And things have improved.

Researcher 16:11

I understand. So. So what other software engineering best practices? Do you do? Do you have code standards? Do you have continuous integrations?

Participant 2 16:22

Yes, we have. We can say we have coding standards, at least in front end, because I work in the front end team as well. We have coding standards, but I will say they are neglected a lot. And not sure why. Yet, at least, because we started the process not long ago. But yeah, we have a coding standards. We also have continuous integration. And there are multiple tools, which which are running like. I'm not finding the word. We have a tool which runs on every build, and it's checking.

Participant 2 17:21

That's automation. I think you mean automation. Yeah. Yeah. There are some some software's which are running, when we ever do a build and is checking if it's fit a couple of rules that it said there.

Researcher 17:37

Okay. Okay, fantastic. So when we understand what quality is, we understand how it's assured and we understand what type of best practices you have. So from the answer, you're provided, now we're gonna get into the details. So from the answers you're provided in the email, it seems like your team or the work environment is highly safe. What do we mean by safe is, it's a work environment where there is a sense of security, there is no repercussions when people admit mistakes. People feel safe and feel comfortable to admit mistakes, they feel it's okay to propose initiative. And there is a sense of confidence that the team will not embarrass or reject or punish someone for speaking up. Do you agree that the team is highly safe work environment? Do you agree?

Participant 2 18:34

Yes, I completely agree. And there were never since I'm working on this project is like now one and a half years they will never let say a tension or reprisal. If someone make a mistake, even if it was if it arrived to the end user. Even then there was no there was no hash let's say there was no no repressible elements right.

Participant 2 19:15

Yeah. So there of course, we sit down and we discussed and how it happened and how can we do it in a way not to do it again. And yeah, there was like, Okay, you did a mistake, but please be careful, more careful next time or in case you need any help just ask someone. Of course, there was like, asking politely to to be paying much more attention in case of a mistake, but nothing too to embarrass or or to feel neglected or to feel bad. So not in that way. And this is why I like very much, and I'm here for five and a half years. So it's really important that person comes, comes to work in the morning, let's say, with a good feeling and not feel bad about he has to work on our software, which he doesn't like.

Researcher 20:18

Okay, fantastic. It's good to hear that. So just put it in a scale. Do you strongly disagree, disagree? Neutral, Agree or strongly agree that your work environment is safe?

Participant 2 20:31

Yeah, I agree. I completely agree.

Researcher 20:35

Okay, fantastic. So the next question is, what made this work environment safe? I mean, from your experience, how the safety has emerged in the team?

Participant 2 20:48

Well, first of all, we have a really great team. And we can talk about anything. Of course, they don't talk about in, in meetings, but they, we talk about in person, but of course, since Corona, we don't really have that much work together as a team, but online only. But we tried to go to the office like two times a week. But yeah, we have a really good team and confidence is, is built day by day. And also, then the new managers, but also the old ones, were really great persons and good to talk to. They encouraged this work culture of safety. And, yeah, we also have flows that kinds of kind of makes the error to be or the issues to be found quickly. So yeah, we had time when it ended to the end customer. But that was like one or two times in a couple of years. But we have flows like it's easy to find someone's issue in an early stage. And then people find not that disturbing to. To correct them.

Researcher 22:17

Yeah, so you've been in this team one year and a half, it has been always like this safe. And and the team is building this confidence, like you said all the time?

Participant 2 22:30

Yes. Mostly, yes. There were ups and downs. And mostly when the old managers were leaving, then it was a bit of tension. Because I don't know why. I would say I don't know why there was some tension there. But together as a team, we, we always discussed if there's something with with them, let's say, but yeah, I would say we were like this is the beginning.

Researcher 23:06

So I sense that the openness of the team has built this safety in addition to the new management, bringing their values. Yeah.

Participant 2 23:17

Yes, I agree. We vary open to each other's. And also what helped that we we sometimes have, like, training sessions with, with professional peoples to build confidence. So we example last summer, we had a course of, I don't know, 28 hours or so with a with a person who's whose job is to, to make the team even more a team and to build confidence and to to say if they have anything, you know, it's it was a training session of, of, I don't know, and more to be open with everyone. So we have session like like this. Like, let's say once a year.

Researcher 24:14

Okay, great. Things. Yeah. Yeah. Fantastic. It's nice to hear that. Let's go to the first question, which I send an email which she says if you make mistakes on your team, it is often held against you and your answer was no. We think like we are human and everybody makes mistakes. We are a team and if someone make mistakes, then it is on the team. And everybody helps to solve it. And we make sure to bring up on the retro to learn and avoid the same mistake next time. We'll you'd like to elaborate in this answer?

Participant 2 25:06

Well, usually there are mistakes, especially because it slips through the testing phase. For whatever reason. I said the testing team is not that high quality. And let's say it, it goes to the system, but not of course, not to production, not to the clients, but it's in the system, which is, everyone is using it for development for everything like that. Even internally, the consultants, everyone, and an issue is found. And it's, it's, it brings up, let's say, in the scrums of we have like three teams. So we have a sponsor of Scrum of scrums. And there, it's mentioned, like there's an issue. And usually, we have a couple of more technical persons who sits there and sits down and investigates where it came. And we found that like, let's see, it's from Team A, and it was introduced like two days ago, and then we just bring up in, in the, in the stand up, and we create a bug or, for example, in our system, we are using Azure DevOps, by the way. We create a bug and then let the person who actually worked on that, okay, here's a problem, and please correct it, and then that person takes over and fixes it. And or even if it's a complex problem, like more difficult to fix, then we sit down together and discuss why maybe it's an architectural issues or it's coming from, not from the architecture part of the of the whole system, then we sit down and discuss how can we resolve it? When we collectively contribute, then we solve difficult problems with better solutions. And, yeah, we'll bring it up directly, usually, if it's a bigger problem, okay.

Researcher 27:27

Do you have a good example to show me how admitting mistakes, this is admitting mistakes, helps quality in any shape or form? So you say that admitting it said it, it helps to get quality? Well, basically.

Participant 2 27:40

I myself don't like when someone tries to hide the mistake, because that could lead into other mistakes. And forever, forever, I can sit there. And that's why I and the whole team is like, Okay, if there's a mistake, then. Yeah, we should tell everyone and pay attention that okay, there's an issue and we are working on it, and we will fix it. And yeah, I think this is a good flow for everyone to and in things, also, knowledge to the other teams that okay, look out their mistake. And maybe that mistake is it has an impact on their work.

Researcher 28:52

So what I understand is, the more mistakes you identifies, as a team you learn from that mistakes, and the repeatability of that mistake is the likelihood that you repeat the same mistakes is, is likely low, right.

Participant 2 29:15

I didn't get the last word because

Researcher 29:18

So the more the more you learn from your mistakes as a team, you avoid in the future to do the same mistakes.

Participant 2 29:28

Yeah, definitely. And that's why we bring up in the retros. Also, when try to, to tackle that. So we don't get make the same mistakes again and again.

Researcher 29:39

Okay, do you have an example where a mistake was brought up in our retro and how did you go about it?

Participant 2 29:46

Yeah, let's, let's see. Well, it was not a coding error. But but another, let's say issue is there. For example, I was mostly doing work, which is not coding related. And I, I bought up that, well, I'm a developer, but I'm not doing any development work. And yeah, I'm, let's say, fixing some really weird bugs, which took me like two weeks to me or the team to figure it out. And then I didn't really do any coding, and I'm sitting there. And, and thinking that okay, what I did today, well, I did almost nothing. And that for two weeks is like, bringing down my, my work, appetite, and yeah, boarded up on the retro, for example, and then the whole team and also the product owners try to try to take that in consideration and ask, for example, other people also to work on issues like I did work, so I can also do development working in the meantime. This is one example. I'm not sure if it's enough.

Researcher 31:21

Do you have a specific example related to quality to code or design or.

Participant 2 31:36

Okay, at the moment, thinking.

Researcher 31:38

Okay, take your time.

Participant 2 32:01

Yeah, maybe it will happen soon. Because not long ago, it was there was an error, which we introduced like a couple of days ago. And another team's team member found it and asked us to, to investigate. And we ate it, it was a real issue, which we have released, like in, let's say, in three weeks or so. And if it goes out, then all the end users will be affected. And I think we will talk about it in the next retro. So I have that example, not sure if it's okay.

Researcher 32:54

Yeah, go ahead. I'd like to hear that example. Yeah.

Participant 2 32:59

So I can also talk about the issue. So we reintroduced an extra parameter for one of our API calls. But that parameter is only is only affecting the current work, it should affect only the current dpi, which we're working on. But since we added that parameter, not only our PBIS, affected, but the whole system that all the clients, so that parameter gets known in that case. So and the end user will get an error, and the actual part of the system will not work at all. So this is something really, really serious. And yeah, I think

we should bring it up in the next retro and yeah, it will be like, I would say two, it's no one's fault, but two together as a team, we need to find a better approach or something to avoid these things in the future.

Researcher 34:15

So you talk about it as a team and the retro right.

Participant 2 34:19

Yeah. Okay. Yeah. And, yeah. Sorry. Yeah, we talk about it together as a team and everyone tries to come up with solutions, how can we how can we tackle that in the future and we choose together the best one.

Researcher 34:42

So do you think that bringing this problem in and the retro which, which it is a problem that introduce a defect in production? So do you think's because feeling safe to do so?

Participant 2 35:03

Yeah, because that safety is there. And because of that, we can easily talk about it. And since no one is is made the so so it's no one's fault, let's say, in that case, everyone can, even if that person knows him did the mistake, but okay, he's a human and he can do mistakes. But let's say it went through the smoke test it went through testing, it went through even a lot of people. So it's not only useful, and he knows it. And he doesn't even think about that, although I've made a mistake, and I'm bad developer and I will get on edge or something. So they don't even think about it. And in that case, we can easily talk and find the solution how to how to fix that in the future.

Researcher 36:05

Okay, fantastic. Great to hear. I'm happy to hear that. Do you think this quality of bringing up problems and discussing them openly because you feel safe? It does help the quality? Because you told me you had a quality problem before? Do you think that this safety that the team has built up and new management help helps you to improve the quality of your work of the software?

Participant 2 36:39

Yeah, I think so. Because then people get even more comfortable and more. They enjoy the work more. And if you enjoy, at least I'm not doing that many mistakes. I'm making mistakes only when I'm hurried. And I'm asked to finish things quickly. So in that case, yeah.

Researcher 37:11

So do that, do I, I understand that the sense of safety people invest more in the quality of their work, they they they put more effort in the quality of their work. That's what you mean.

Participant 2 37:27

Yeah, I think so. As, I said people are comfortable to admit mistakes and learn how to avoid them.

Researcher 37:30

Okay, I'm looking at the question. So for example, this safe work environment, it's encourage as people to report defects and errors from your experience, right? Yeah. So they have an example? When did you observe that when the new management came in and start promoting the safety? Do you think that people talk more about defects and problems of quality related?

Participant 2 38:09

Yes, and I want to say why? Because in the previous so before the change, we also tried to bring up the issues, the errors, the the problems with the flow. And but they said that, okay, we know about it, and that's it, and they don't even considered to change anything or to, to bring in something to change that. And after a while, you know, people are like, Okay, I'm not saying this again, because I said a lot of times and I won't say it again. So, we kind of stopped saying where the where we saw something or reporting something. And that was brought back by the new people. Of course, because they are new and we are trying it again, but this time we kind of our our sentences arrive to people and they do really listen. Example, we had a new backhander not long ago. And she was coming from another company exactly because she was kind of left alone. So she didn't have any other people to work on or to work with so and she didn't feel that good. So she left and we hired another developer and then suddenly, he started working he received some training and he had another friend working with her. And suddenly another team backend left and she was put into that team. Completely. Strange team stranger, half Lexi, not strange, team. But team, which she didn't know. So it was. And also, she worked from Romania and the other team members vote from another country, and they didn't have any interaction only online, not as the other team members, which we have here. And there, I said to the manager, like, okay, but look, this is why she left the other company. And now we are doing with her the same thing of feeling alone. And in a new team, and I really don't want to, to make her to leave us as well. And he said, like, okay, yeah, thank you for saying this. This is a really good observation. And we will keep that in mind. And yeah, they really did steps to, to, not to to make this happen. So they really hear what I said. And

Researcher 41:30

Yeah, I understand. So they listen. And they, they took the right action, the corrective action, so it's encouraging to keep bringing problems. So that brings me to the next questions, which I sent before. member of your team can bring problems and tough issues? And you said yes, in every team that are problems, issues. If it's a management issue, we try to step it up and say, to the higher management or to the higher person, if it's something we can resolve, we resolve it ourselves. So do you have a problem related to software quality, where somebody brought up a problem, and you talk about it as a team or raise it to management? And it did help to improve your ability to achieve better quality?

Participant 2 42:24

Can you repeat it? Because you were interrupting in the middle of your sentence? Oh, okay.

Researcher 42:29

Sorry. When I was interrupted, so you said, the question is member of your team can bring up problems and tough issues? Yeah. So your answer was yes. Well, in every team, that our problem issue, it's a management issue, we try to step up and say to higher management, if it's something we can resolve

will resolve it ourselves. Right? Yeah. Yeah. Fantastic. So do you have an example where a member of the team brought a problem to the team or to the higher management, and the problem was dealt with, and it has helped the team to achieve better software quality?

Participant 2 43:21

Yeah, for example, our UX or worked in a software. I don't know the name, where we were, he could do the designs and do prototypes and stuff. But that was not really suitable for him. And then he asked the team, okay, is it good what I deliver to you, because I feel that what I delivered to you, as developers, it's not that usable, and maybe we should find another software to do the prototypes. And yeah, we sit down and said to him, but okay, we are using your, what you are sending to us in this, this and that way. But yeah, it's a lot of work to to take out what we actually need from your designs. And then we found at least he proposed another software, which he used in the previous company, he was working, and we checked together and yeah, it was like a better, better one. And it gave us more value to developers for how the design and the prototype like direction is. And then we try to take it to our manager and said that okay, what do you I think we should switch from this software to that software because ABC. And yeah, we know that maybe it's, it's more expensive. But okay, this is the upside. And this is why we should use that one and not the only one. And please consider if it fits the budget if we can do it and accept, for example,

Researcher 45:28

Okay, fantastic. That's a good example. And it's bringing me to the next items, which wasn't my email people on your team sometimes reject other for the indifferent. In here, what we mean by rejection is good for the example you used. People brings idea people brings initiative, and they don't get rejected based on the initiative. So that's what we mean by rejected. So we never reject a person. That's what you said, if someone is different than the rest of the team, and it does not fit, then we try to change things, and we find the best place for them and get the best out of it. So in that example, that you use, the person took the initiative and propose an alternative. And he was well received, right? Yes. Why he why he took initiative in the first place, in your opinion.

Participant 2 46:32

Yeah, it was a person who I felt like daily basis, and it's a friend of mine. And at one day, he he asked me like, okay, but is it? Is it fine what I deliver? Because I don't really think that this is good. And I'm not sure why he did. Maybe he, he's also a kind of perfectionist, so he he really wants to deliver something that it's, it's really good. And maybe that's why he asked me. But also because we are good friends. And we talk about basically anything. And yeah, because of the old managers. We sometimes also made jokes of things like, okay, but jokes like, you know, which we can make, because it was so bad, but we can joke about it still. So it's let's say it's a bad flow, but we joke about it to make it better. Maybe because of that he brought it up.

Researcher 47:57

Okay. When you changed the new software, did it help? Somehow the UI design quality?

Participant 2 48:10

Yes, yes, it helped a lot. And yeah, it's okay, that person already left for another company, but we really missed him to bring the next initiatives like this, it was very proactive and yeah, it is what it is.

Researcher 48:30

Okay. We will move to the next one, which is about helping each other. So, I asked you, it is difficult to ask other member of your team to help. And you say no, we are a team and we constantly help each other out. We are even we even take other roles if needed and do their job. If they are busy or unavailable for some reason, in most cases wouldn't even ask. So this quality of helping each other. Why do you think it has emerged in the team? What's brought it up in the team?

Participant 2 49:17

Good question. And the answer is at least for our team is because the testing team was not that performing and we had a lot of issues testing and they were also behind in schedule. So like we had 10 new stuff developed but none of them tested and in that case, one of the team members came with the idea okay, but doing this further is it's useless because then we will never release anything because the testing team is far behind and we are just adding more and more and more work. So let's try to help them somehow. And we asked the testers How can we help and we agree on on a process which can help them and we simply sit down and we were testers so everyone became a tester, but of course in a way that not to jeopardise the quality. So they, they set it up, let's say all the test cases and the other team members who were developers ran the test cases, because they had a good knowledge of the system. So they will, it will, it was easy to run the test cases which are in front of you. And at the end of the last words, of course, the senior tester had set the goal, okay, this is green, really, it can go. And this way we, we stopped development for, I don't know, three weeks or something like that we finished all the test work. And then we tried even to make a new process, which makes not to make the same mistakes, because it will end up let's say, in a year, we end up in the same situation of the test team lagging behind. So we also made try to meet at least, and it's working, at least for the moment, we made a new flow or process. And Scrum. Scrum really helps this because we are chunking. The we make really small chunks of work. And we develop it, we test it and we go to the next chunk and next chunk. And this way, we don't lag behind with any of the or the team.

Researcher 52:08

Yeah, thanks for sharing that example. It's a good example. Do you think sharing the knowledge with the testing team and bring their knowledge to the same level as the developer teams has helped improve the quality?

Participant 2 52:29

Not only the quality, but it helped us to actually deliver and, and give, give the value that end user needs.

Researcher 53:08

So how did you deliver value?

Participant 2 53:14

Well, it had quality but not the maybe not the top quality we wanted. That's why we are looking for, of course, new testers to make that thing bigger because the lack of resources is also a big problem. That was that was the main problem.

Researcher 53:41

So it didn't help to the level you wanted to help. But it did help to some extent, right? Yeah. There are two items I'd like to go to. No one on my team would deliberately act in a way that undermine my effort. So yes, true. This is a great quality. Yes, we are good team and we are working on the same goal to deliver and finish the sprint can give a value to the business. No one tried to undermine one effort. This is really fantastic. For example, let's work with this scenario. A developer who decided to invest extra time to do a good quality code. Would his effort be undermined by the team for example?

Participant 2 54:42

No. And even others, if they see that he succeeded, even others would adopt the same thing. So we have like maybe this is not related. We have like, a weekly of two, three hours to free time to develop ourselves to be even better. And in that case, we can buy courses we can learn, and the new stuff because of course, we have some legacy, we try to keep it with the new stuff, but we do have some legacy. So there are two or three hours weekly when a person can make himself better.

Researcher 55:37

Okay, fantastic. Is this something that the management promoted? Or? Okay, fantastic. How does it make you feel when you are given time to invest in yourself to become a better developer?

Participant 2 55:54

Yes. Everyone likes it, because it it, it gives you the chance to to develop even more. And yeah, you know, we have we are can developers are said like they are kind of nerds. So they really like to, to check what's what's on all around the world. So what's new? Try stuff. Sometimes we even sit together and and do, let's say a minor project. Right? Yes, exactly. We are trying since years to to change the front end framework, for example. And yeah, we sometimes sit together and try to, to explore React, for example, because we know that that's really popular now. And we want to check if it would fit our project or not.

Researcher 56:57

So this this, this desire to experiment and try. Do you think it's because you feel safe in this team?

Participant 2 57:09

Yeah, yeah. It's related. Yes.

Researcher 57:15

Okay. And when you experiment, when you explore and you learn something, does it help with the quality of your work?

Participant 2 57:25

Well, it depends on what Yes, of course, because with the example I said of the React for now, it doesn't have any impact on our we just learn another framework, which we might use. So, there there is no really quality added for now at least. But yeah, there are situations when it helps.

Researcher 57:52

Yeah. So, if you had to learn a static analyzer, for example, if you wanted to learn a static analyzer, you would feel safe to do so and you will even take initiative to propose it to the team right. So, the same thing applied to a quality process, nothing would stop you from doing it. There is the last question which is says working with a member of my team my unique skills and talents are value and utilise. And you say yes, my talent and skills are utilise, and also other skills are gained. Because we constantly help each other out and learn day by day. Of course, everyone has its main role, but if needed, we can cross our work. That's fantastic. I mean, when you work in a team like this, you cross function and you help each other. What's what's qualities is bringing to the team I mean

Participant 2 59:10

Well, yeah, we do cross work, then of course, that means a person who is helping out for example, I as a front end, or if I help out the tester, that means that I will get better and better right in, for example, in testing. So this is how other team members have better quality of can do better quality.

Researcher 59:40

Okay, fantastic. So we come to an end we have one minutes. Is there any things you'd like to add about this feeling safe in your team and how does it help the teams to deliver better quality?

Participant 2 59:57

I didn't get the last

Researcher 1:00:00

So what I wanted to ask is, if you'd like to add anything that is related to this feeling of safety and the work environment and achieving better quality or improving the quality processes you have in the team.

Participant 2 1:00:19

Well, in addition, I don't know, it feels really good, and I enjoy what I'm doing. So that's what I want to add. And it's really good to have a team like that.

Researcher 1:00:38

So when you enjoy what you're doing, do you tend to, to invest more in the quality of your work?

Participant 2 1:00:47

Oh, yeah, always. Okay.

Researcher 1:00:50

Okay, fantastic. I think we should conclude here. We will, what we use, I will be talking to a lot of software developers, and we analyse this, and we will come with some inclusions. And I'm thinking to

organise a focus group sometimes in June to discuss some of these findings. Would you like to participate? Yes, I can send you the invite.

Participant 2 1:01:40

Yeah. Yeah, sure. Sure. Okay. Fantastic. Available. Yeah. Yeah, of course. If you available. Yeah. Yeah. Yeah. Okay.

Researcher 1:01:53

I'd like to thank you again for doing the interview. I really appreciate and thanks for the when it's Saturday morning. I enjoyed the discussion. And I wish you a good weekend. Bye.

Participant 2 1:02:10

Yes, thank you very much. Bye bye. nice weekend.