Participant 14

SUMMARY KEYWORDS
people, team, code, working, test cases, quality, test, software, helps, applications, learning, running, big, problems, developing, write, person, important, improve

SPEAKERS
Participant 14
Researcher

Researcher 00:16
Hi, How are you?

Participant 14 00:18
Hello. I'm good. Hi, Researcher.

Researcher 00:20
I'm doing very well. Thank you. Where are you in Sweden?

Participant 14 00:24
I'm in Gothenburg.

Researcher 00:26
I have been to Gothenburg. It's a beautiful city.

Participant 14 00:33
I've been to Copenhagen in Denmark.

Researcher 00:35
I mean, all I mean, all by far north. Okay. Okay. You work for [deleted to preserve the researcher/participant anonymity] right?

Participant 14 00:45
Yes. My parent company is [deleted to preserve the researcher/participant anonymity].

Researcher 00:50
I worked with the research branch of [deleted to preserve the researcher/participant anonymity] in my PhD project. Yeah. Anyway, I'd like to thank you for accepting to the interview. And I appreciate I do have a lot of questions. So let's start. Do you have any questions for me before we start? All right, fantastic. Perhaps a brief introduction. Just introduce yourself, your education and your experience.

Participant 14 01:25
Yes, so I graduated as a computer engineer from [deleted to preserve the researcher/participant anonymity] 2014 I graduated, and then I started working as a software developer. One half year, one, around one and a half year I've worked [deleted to preserve the researcher/participant anonymity] in started with software as a service aged payment processing. So it was FinTech mostly in the US. So I've worked with like, did some development for big clients, like the [deleted to preserve the researcher/participant anonymity] and they were [deleted to preserve the researcher/participant anonymity]. So it was a really big company, you know, really tight deadlines. So I got a lot of exposure on how this off really works, you know, first time moving out of our small university projects, real stuff. And then after one year, then I came to Sweden, I got a job offer via LinkedIn and stuff. So they're, with something completely different from a big large corporation to accomplish very small startup, you know, a lot of freedom. And I was doing previously it was Java, but this was dotnet. So it was again, something different experience from me, learning new stuff, and it was more freedom, I meant that we didn't have any processes, you could take the lead on basically anything. So I learned a lot there that we had this migration stuff like moving this, you know, first prototype of a startup into some real big deal. So it's got experience with cloud computing, you know, migrating everything to Amazon Web Services and dividing this monolith into micro services. So a lot of us involved, lots of decision making, and it's just full of innovations, you know, do whatever. So got experience with that. Again, it was software development and veneer there then I just finished my Master's in Computer Science at Chalmers. So I started studying again, a master's in computer science with specialisation machine learning about this in this second, my startup company for one year. Then I joined my present company everywhere, I've been working for three years. And my first assignment is consultancy. So my first assignment was [deleted to preserve the researcher/participant anonymity] [deleted to preserve the researcher/participant anonymity], where I worked from March 2019 until Corona started. And then it was falling out the you know, it was [deleted to preserve the researcher/participant anonymity] group connected solutions. It was like really big domain. And, you know, there were a lot of processes, they use a Scaled Agile Framework, not a textbook version of it, but their own interpretation of it. So, I was really exposed to this like read they have this real machinery that you have to follow these things and their meetings that they have their retrospectives on their time, they plan for two, they pair for 10 weeks. So five Sprint's in advance like this is done, you have learned about this 20 different teams that you have to collaborate with, you know, your everything is delivering and there's all those dependencies involved. So you need to sort them out what is needed first as you can do things, you know, for the 10 weeks that this thing cannot be done until you deliver this a you need to plan and sort everything. So you have worked for the next 10 weeks. So a lot of got a lot of exposures, engage Island stuff, and I also had some experience as a scrum master. When my like scrum master was on vacation. So I took the initiative to learn this thing. But I was working as a Java developer again. We had some components not normally, previous at the beginning, they gave us small tasks like management, moving migrations, migrating systems. But then I got more responsibility and bit deciding on how things develop to support the company, we were developing a lot of micro services in Java that run on the AWS and they were like the core of [deleted to preserve the researcher/participant anonymity] group connected solutions. And so all their vehicles were coming in our applications and you're managing everything, you know, this truck is there and it's offline and online, you don't they have disconnected trucks, they had the SIM card, they tell them all the status and everything. So it was a lot of load, I was like managing million that it got like a million status updates per day, you have to run migrations without

taking down the system. You know, it was a lot of new things, a very hectic thing, but was a lot of collaboration was involved. We were doing talking to us. So UK office, India office and Gothenburg office and at this time, so I learned a lot of this soft skills management and everything. And apart from that is a lot of development experience. Managing these big systems, you can't take down a [deleted to preserve the researcher/participant anonymity] system, you know, is taking down a startup system using real estate stuff is not much but if [deleted to preserve the researcher/participant anonymity] system goes down, it's a real big deal.

Participant 14 06:18
Then, after that, when working now with a free [deleted to preserve the researcher/participant anonymity] for one and a half year we'll dissect again a different switch for me with developed like [deleted to preserve the researcher/participant anonymity] is working on this IoT industry 4.2 kind of do great feel idea that is working for developing for factories. You know, like robots is kind of a defy when robots gonna fail, and they're gonna make before it fails decisions, you know, so if it's gonna need be reading oil in two days is too late former as before, as soon as possible. So it's kind of like that at the end of it. So here I was developing everything from scratch. And they're also where we are like, people have six 617 67 people in a team. And there we have several teams across different domains, business applications, platforms, and you know, a lot of visualisation graphs. So team, we have people divided across the teams, and we are developing everything from scratch. So we have developed, basically now it's technology shift for me, I'm working with TypeScript and Node js and react basically applications. It's again, Docker, and you know, those DevOps things, but the difference is that they're gonna be running on a tiny IoT Edge device in the factory, probably not even connected to the internet. So it has been I've been, I've experienced a lot of different experiences, everything I have is five years, but I probably have five experiences of one year each.

Researcher 07:53
I'm happy to hear that. That's you having fun, which is good.

Participant 14 07:58
I like learning a lot. And I just finished my master's as well. So it's like to end of December.

Researcher 08:05
Okay, congratulations. Thank you. The answer. Thanks for sending those answering the email. The answer you provided are based on your current team.

Participant 14 08:17
Yes, I think the vast majority is probably on my current team.

Researcher 08:21
Okay, fantastic. So I will ask you a few questions about the team and we will go to those answers afterward. So are you using Agile at the moment? Right? Yeah. What type of Agile method you're using?

Participant 14 08:39
We've been working with scum. Okay. So exclusively for the past two years, I think.

Researcher 08:45
Okay, fantastic. So, how big is the team?

Participant 14 08:54
I mean, it's some people work like, you know, It grows and gums but normally is I think it's six, seven people. Normally, it hasn't grown up, but he says sometimes someone moves on to other projects, you know, sometimes 50%.

Researcher 09:13
So what type of software? I assume you're developing software for robots?

Participant 14 09:21
Yeah. Right. Yeah, the more software parts we don't deal directly with robots is they come they send their messages and we are on the part that is processed those messages. Yeah.

Researcher 09:30
Yeah. I understand. Is it is it a new software or I assume with [deleted to preserve the researcher/participant anonymity] is an existing software right.

Participant 14 09:41
Now the system's developing work is they're building from ground up.

Researcher 09:47
okay, okay. Okay. Last question, and we will move to the core part of the interview. How self manage Are you how much control over the team is there. Also, how long have you been working together as a team?

Participant 14 10:03
Since I joined one year and a half. I would say we are very independent. We have also because we are also very report we are in one Sweden, we are in Czech and we are in Spain. So, it's almost completely independent ways. Just like when every two sprints we have demos and we say that this is what we have done. And then we have only the scrum master mostly is involved in meetings where like what needs to be done and prioritising otherwise, splitting the planet apart from customer demands, but the stakeholders want create, develop a lot of stuff ourselves to solve the problems that we encounter.

Researcher 10:42
Okay, great. We will be talking about examples regarding software quality, and we will discuss your answer in relation to software quality. But as you know, software quality can have different definition, and people can have different opinion and perspective on it. So I'd like to confirm the definition we're using. I'll read it to you. And we can discuss this briefly before we move to the rest of the question. So we use ISO definition, and I'll read it to you. So it says software quality is the degree to which the

system satisfies the stated or implied needs of its various stakeholders. And does provide value. The ISO model related to quality also cover some non functional characteristic like performance, compatibility, usability, reliability, security, maintainability, and portability. So first, what do you think? or to what degree do you agree with this definition? or disagree? And would you like to comment on it?

Participant 14 11:55
Yeah, I think I mean, it's right software quality is all these things, performance are so important for us, since we are running on a small device. And of course, I think the stakeholders, whatever they come is always a top priority. If someone's returning test many tests in a factory, we always support them. And also, it's important to be able to have reproducible environments for us. So we can even if it works in dev, it should work in production, you know, so we have several environments. So all these things are things are gathered around of satisfying this definition. You know, like, we have our unit tests and integration tests, tests, the purpose of them code reviews, and we think all we do that part from that we also focus a bit on code quality, you know, people are gonna collaborate across teams, you know, so badly written code, poorly formatting things is also something that is important for us. And I think we also consider that as part of software quality.

Researcher 12:55
Yeah, I agree with you. Thank you. So what processes do you have in place to assure this quality?

Participant 14 13:05
Yes, so we were to use very using a model continuous deployment. So it's, you we have different branches. So branches, environments are connected, we you push, a push to dev branch code gets deployed in all the all dev environments, you know, every team has their own dev server. If you push to from dev, you have to merge to staging. Okay, let's start like this. I have a you have a feature and a feature bug, you know, we're gonna have someone's gonna report it. They're using Microsoft Azure, DevOps to track everything, you know, it's like JIRA. But it's everything is Azure DevOps. So stories are there, they're groomed. And you know, they're prioritised. Because like, say, the stakeholders, what they want is the most priority, but if there is critical bugs as well, then, you know, there's also go up on priority. So if a bug is there, we make a branch in the code, we're going to normally if I'm developing my service, if I have a bug, and I'm very experienced with my service, I'm probably gonna write some unit tests and test first, to make sure that I can reproduce the bug in my system. And if it's there, then that is the part to write the code to actually fix it. There's the testable staying is sometimes just not possible here. So many things, machinery involved it as all trading tests are not always possible, you know, like, HTTPS is not there in local environment testing some problem because of because of HTTPS is not so easy to write a small unit test. But once we have done test I've tested locally so we're using Docker so make sure that the compiled code is actually the same thing that will be run in production. So we do everything in Docker environment. So we use Docker Compose, so we spin up the entire stack that we have on our machines, you know, the database is running and everything is there and do our Last year, and once we are satisfied, then we merged to tf. As soon as we must adapt the pipeline runs in Azure DevOps automatically. And it's also running the tests again, you know, the automatic tests, once they run, then the Docker images get built, built, and then they get deployed to the dev environment, the dev environment is then we go there, check the logs in the backend, you know, check the GUI that everything is working, you know, that's normally sufficient. And then we move to staging. Staging is

supposed to be like, everyone's ready to release components are there, we only have one staging, you know, so my team components are there, the team's component is there, everything is there. And that's where we all do also do our demos, there's like just the stop before release. And that is very important to make sure that everything is working together, you know, in because in my dev environment, okay, I did something is working, but maybe someone else is working on a component, and then both get released, things just go kaboom, you know. So it's important to do it in staging. And once this is done in staging, then release is kind of you need to get and give a heads up in our scenario, because these IoT devices, they're probably hundreds of them as soon as we publish a release, and it just goes right now it just goes wild. But then we are there. We just published a release. And normally, we expect there no the problems here, but there still happens that in production that problems come and then they have to follow this same cycle over again, you know, Dev, staging and release.

Researcher 16:39
Okay, great. Thanks, Participant 14. I appreciate let's move to the core aspect of the interview. And based on the answers you provided. My assessment is you work in a team which is highly safe work environment, I will explain what do we mean by safe, so safe in this context, it means a work environment wish people or the team member believe and feel it's okay to admit mistake, it's okay to report mistake. And they believe and feel it's okay to accept and discuss problems. They also take initiative and they feel it's okay to ask when they need help. So we i Based on this definition of safe, I think the work environment is highly safe. Do you agree with me? Would you like

Participant 14 17:39
I completely agree what you just stated?

Researcher 17:42
Okay, great, too. So in a scale of five, strongly disagree, disagree, neutral agree, strongly agree? Which one do you agree will you would select? So would you strongly agree or agree or neutral disagree or strongly disagree? For which, okay, so we say it's highly safe work environment. Right.

Participant 14 18:16
Okay. On this statement? Yeah, I would say it's five.

Researcher 18:19
Okay, strongly agree, right. Yeah. So what makes this environment highly safe so what happened or what's the things that make this work environment with this quality?

Participant 14 18:34
I think the most important and since the most I enter interact with is the team I worked with. I guess is the most important thing is the relation that you have there I have with my teammates. Now we're working for the past three years, you know, we started working when we never met each other. We met like five minutes before the interview for the assignment. And it's the thing I think is the human it is the contact that has developed like you know, going out together having the team activities together, having eating lunch together it kind of is when I approached my team it's not like we approaching just a colleague you know don't get hurt never have this feeling you just talking to a friend you know, because

I know the feeling that okay, i Here's all you need for this issue. You need to talk to this guy and never heard their name. As you write to on Microsoft Teams, you have this different feeling like you know who they are, and you know, that they're busy, you know, you have this question, but it never happens with them, I think is just a thing of having working together and knowing each other as a person, you know, being helpful and that's what has developed this culture.

Researcher 19:47
Okay, great, thank you. I'll move to the to the answers you provided in the email, and we'll discuss one by one and if you can Share some example would be great. So the first one says, if you make mistakes in your team, it is often held against you. So your answer was no, we have always shared the responsibility, and do not blame each other in the team, we focus on how to improve as a team. So can you comment in this quality of not blaming?

Participant 14 20:27
Yeah, I mean, it's, it has happened personally with me a couple of times, we were working in this Falvo system. And we had this same thing migration happening, there was a US UK office and in the office involved, and we had this protocol, you know, we had this, which was financing, billing, relating stuff. And we had this first we're gonna do this, we're gonna do that. And that, you know, it was a lot of going on, and I, I kind of deleted the database to clear to move to the next step while they were testing. And we were just not talking to each other that time. And that kind of at the same time, everything broke for them. So they were completely like, but what has happened, you know, so they were not, of course, happy with that, that, you know, we decided to delete the entire database without a backup. So, yeah, I'm in the next meeting. It was they were not, of course, very happy. But yeah, my scrum master, kind of my team lead joy that we just say, Oh, we are very sorry that this happened, you know, this, these words, you know, and yeah, we'll be, we will take notice of this and tried to make sure that this doesn't happen again. So for me, I felt myself with, you know, I should have followed up, you know, I gave them a heads up. But so we have like daily meeting set nine in the morning and stuff and all retrospective and they say, yeah, sorry, this is, this is my fault. You know, it was there. They're just not happy. But I mean, there were no consequences. But you know, they were just not happy. So but, of course, you don’t need to take the blame on yourself instead we take it as a team, because we can recover from those things faster. So then I felt like, yeah, it's, if I make a mistake, it's nobody gonna point fingers at me, you know, if we go their way, if they put a plate, they put up them on a diet, if that's the name of a team, they may name something they're not. Nobody ever says that, you know, that this person, you know, it's like, okay, this, we have this problem this team is, Participant 14 is responsible, they approach them, and they will help you fix this. So yeah.

Researcher 22:37
Okay. That's, that's a great example. Thank you. How did it make you feel when the whole team took the blame? It was not a blame on you.

Participant 14 22:46
Yeah, I mean, I, I've just, you feel happy that you've worked with people that you can rely on. So probably, I'm working with the same people for three years now. And I don't want to leave them. Otherwise, previous companies I've got one year. So I think that tells how I feel about them.

Researcher 23:09
So in the individual level, what did you learn from this mistake? Did it influence the way you work?

Participant 14 23:21
It wasn't made that previously, you know, most of development work has been like more independent, and I do stuff and they say, but this this was more collaborative. So it's kind of not that you have and you need a bit better than medication when you're doing a lot of this synchronised operations, you know, so it's important to know what we agreed on what we will do when in what order? There's some degree it was a lesson for me that Yeah.

Researcher 23:48
Okay. We will move to the next item on the email. It says a member of your team can bring up problem and tough issues. And your answer is yes, we have a retrospective after every sprint, where we can highlight the issue. I work in Sweden, where people avoid conflict. So is unnamed anyone in the team but convey our thoughts in a way that does not point to anyone? The retrospective only includes the team member and no one else? That's very interesting, and healthy way of approaching it. And why do you think in this Swedish context, you don't single nobody? So when you bring up an issue, it's it's agnostic of the individual. Why do you think you do that?

Participant 14 24:51
I think it's important thing in team that, that we as a team, you know, it's, we're not trying to be better than each other, we are trying to be better as an entire team, you know, so people can take it, you know, some things personally and if something is wrong, maybe then they are already aware of it. And you know, maybe it's their feeling about it, that there's something wrong with them. And there might be the demotivated because of that. So we don't need to point fingers at them in public, they just alleviate their regional feeling. So it's better to tell them that we're here to support you, and we will grow together. And if you have a weakness, you can improve it, you know, so I think it is very important. And I think that's the secret of one secret of team succeed, you know, there is not for each individually is all together, you know, as you say, the chain is as strong as its weakest link. So if, if we have, we know, there is a weakness in the team, we help them improve it. So, no, to make the weakest thing as strong as possible. So, and Swedish culture, yeah, it is itself is I think, it helps it this that they don't, they avoid conflicts, like if they don't point fingers at someone, then they they're very afraid that they might come. Now, they might fight back and argue the Finnish Swedish people, they will start shaking if they hear so many arguments or loud voices. Loud noises, it can be bad, sometimes the bad effects of it, but I think it's helps as well. You know.

Researcher 26:28
That's why they are a neutral country, they never go to war.

Participant 14 26:30
Except we leave it to Denmark to invade the whole Scandinavia.

Researcher 26:38
Thank you. Can you share with me an example that in a retrospective somebody raise a team member raise a quality related issue? And how did you go about it and the retrospective?

Participant 14 26:59
I can say that first example comes in my mind is it I saw that you have you have some services, and their tests were written, but they're not working. So they're useless, you know, they're outdated. And I was first time a second time developing in a service. So it kind of felt hard to test anything, I'm making a change. And I don't know, what's gonna happen, you know, is a good critical components. So I kind of raised the point that, you know, first of all, the code is very badly formatted, I cannot look at the brackets and see where it's starting in various ways this function is ending. And then what's the point of having these test cases if we can't test it? So I kind of probably went a bit harsh as well. Because it has been bugging me for a lot. So yeah, I told that it was within our teams, you know, so it's even more hard. You can blame other people from other teams. But this was within our team. So that's what I did. And, of course, the person who has been working with it a lot understood now, but like I said, previously, I didn't blame him, I just say this component is important, you know, and we should do it. And there's a whole responsibility. So the person who worked on it will then later approach out. Yeah, I have these cases. I know they're not working. But I'm not really so experienced with them, I don't know how to improve them. So we'll need your help. You know, because I've been doing more test cases I've experienced with the variety of integration tests, then they looked out. So we just had a meeting, like, take a look at this, this is a really big work. It's not like, you know, I thought that maybe they're just not updated. But it seems it's is going to require a lot of work to fix them. So kind of I got a piece that, you know, this is not a simple task, we're going to plan it is going to probably take one sprint. So that's how at least raise an issue. We know normally we do have other issues, like if my scrum master, like the previous sprint was not so to get too big to her hard on him that yeah, we couldn't accomplish 100% of the objectives we agreed on. You know, she was like, Oh, this is not nice. But what should we do with then everyone was like one person, like, you know, I've been working 20 years this happens, and I think need to take it so seriously, you know, nobody questions you like the stakeholders never say, Oh, you didn't deliver everything you asked for. So and of course, why did this happen? Well, yeah, he had sickness. One person might Corona and so he couldn't do it. So it was like, relaxed the situation if someone is just feeling tense, that you know, this thing is probably cause a problem, whereas it doesn't. So the these are very tough conversations to help, you know, because some people might think that you know, someone is sick, they might think that oh, we couldn't achieve the goal because of me, you know. So doesn't it explain the damage if you're sick by the not to be blamed for it, you know,

Researcher 30:02
I like the example of the test cases and the coverage of the test cases. So you helped your, your, your team member to, to improve the test cases, right? Yeah. Yeah. So what what's the what the advantages you achieved from that? So the the test cases got better. So do you think that the quality of the code that the test cases cover that feature gets better?

Participant 14 30:34
Yeah, I mean, the thing with quality is that a normally that's we're writing test cases, for the part that's already working, you know, sometimes you're doing that we're not doing full TDD, you know, it's not always practical. But what but this allows with a test case, test cases, first thing, when you're writing the test cases, you kind of face problems in your code, because your code is not testable. So you have to make refactoring. So that improves the quality on its own, you know, follow the solid principles a bit more to make it testable. The second thing happens is next time, when you have to do something, someone else has to touch the code, having those test cases help in that case, because you make your change. And then you read run your test cases, and one minute, you can figure out that, okay, the core functionality has not broken. But if yours, there is no test cases, then you're like, Okay, I have changed this one knob. What's gonna happen? No one knows. I don't even know the service. I don't know what to test. So they help you a lot, you know, you use like an investment. You know, it pays off the next time you work on it.

Researcher 31:38
Yeah. And it's assured the maintainability and the evolution of the software. Yes, yeah. I agree with you. That's a good example. Thank you. We'll move to the next one, which is about rejection. people on your team sometimes reject other for being different. I read your your answer, and we will discuss it No. In my team in my current team, that I have been working for around three years, no one ever gets rejected or gets personally attack. We do have conflict, but we attacked idea, not people in a respectable manner. I like that we attack idea but not the people. It goes like this, someone say, Oh, what if we do such things? And other people say yeah, but what would happen if such thing happened? So as a team you discuss, and you make your own decision, and you make your own decision how you go about things, right? This is about what I understood.

Participant 14 32:46
Yeah, I mean, we, of course, we have the bigger purpose of the objective, but we are very open to having the technology to technology is now being set, but how to solve a lot of problems, our stakeholders mostly come to us with problems and rely on us to offer them solutions, not not tell us that you need to do this, and you have this much time. So that's why it allows us to be more flexible in this. Adding on this, we do have differences pair programme with one of my colleagues on a project, even from [deleted to preserve the researcher/participant anonymity] and even then, we were pair programming for five hours a day, you know, so it brings up differences. I'm more a bit more very to having this code Linters and code from eating and everything set properly, you know, and really to code for meta structure and spelling errors I don't like and by my my colleague, on the other hand is a completely different guide, it works the No need to touch it, you know, so we had our differences. And yeah, I was like, Oh, why you need to do this and then like, just do it, you know? So but sometimes I felt the idea that he doesn't like me doing that, but then we have this team activity and then we were talking and then he told ya even he works with me. I know I don't like any of this thing is just a waste of time. You know, this is the idea is always crying usually have an extra space. But he says he still lets me do it. And he still trusts me that you know, in the end, it's I don't like it but when the doula does it and he is good that because in the end, it is helpful for the next person to work on it, you know. So even though we had a differences, you know, we don't have to go advantages now. We have automated tools. So I

wrote a tool that when you eat farmers to code CTRL S when you press ctrl S it doesn't ask you for your opinion, so you don't have much choice anyway.

Researcher 34:48
So this is a good example I mean, your style of work. This is what we mean by rejection is having two different style and views of work, but You did the class, you, still collaborate together, right?

Participant 14 35:04
Yeah, because we had this team activity, we had this, one of our managers, he's into these people and culture stuff, you know. So she asked, she has these things, I don't know, it's always this pace in the red box, you know, these kinds of things, but kind of your thinking processes. And she does these exercises with all of us. And then she tells each of our personalities, you know, like, Oh, your is more concerned about this, you know, is more so concerned about that, you know, everyone has different priorities. And she can tell our, she can tell how what kind of person we are by looking at our, our computer's desktop, some people have this cluttered icon, some have different colour organised, she's even factor that into account. So it does that. For a team to function, he had all these colours, you need all these colours with some quantity, if everyone's red, the team's not gonna function, you know, if everyone's yellow, then it's not gonna affect you, you need a spectrum, you know, like a rainbow, everything. And that's, that helps you in that series, as your team has this much percentage of this. So it's a healthy team. She said that a long time ago. And she even looked at it, and told us that if you have a meeting going at one, at what point of time, will what? Which one of us will have what kind of response? You know, is the logo, this introvert person? Now he's going to speak very late in the meeting, but you have to let him speak. Because if he's going to try to speak couple of times, if you're not let him is just going to stop. But I can tell you he will have the answers. You know, it was like these things. So I think from that we kind of appreciated our differences as well.

Researcher 36:48
Yes, there is a lot of research going onto that. And it shows diversity brings strength to a team. Yes, I agree with you. Thanks for sharing that. I'll move to the next item, which is about taking risk and initiative. So it takes it says it is safe to take risk and initiative in your team. You said yes, in my current team initiative outcome. And I have developed a number of things without any clear requirement and demand from your stakeholders. The important thing is that the things we commit to the stakeholders are completed, it was as a team to commit and we leave some room for such things. So can you share with me an example where yourself or a team member took an initiative related to quality? And how do you go about it?

Participant 14 37:43
In terms of circularity, personally, I have developed the thing is that I developed an basically an identity and access management service signing stuff, you know, you do signings using Google. So I developed it, and they said, Oh, we want this, you know, they come with this requirement, we need to be able to sign in, you want this thing, you know, and the architect came up with, I think this is this is my research that pages take a look. So again, I made the survey and stuff. And I was like, yeah, it seems a bit complicated for people, for me to explain to people how to integrate it. So yeah, I have the time, you know, sometimes I just ask for more time normally says gonna take time, because I know that I need to

do my things, you know. So I developed that, but then apart from that, I developed a front end client, so the front end people can integrate to it. And then I added a back client back end library. So the backend, people can integrate it to make make it easier for their part of the code. And on top of that, and to help people because I know people are gonna come to me every day, if I if I attempt to give them this is a complicated solution. So given given the time, it was a bit summertime, you know, a bit of a job. Not many people working, whereas given the opportunity, I made sample clients, I made sample applications, which have to code and everything was like, Yeah, this is how it works. I explained to them, and you just look at the sample applications, you know. So it was a lot of I think half of the stuff that was I was never asked to do. It's something I thought that this is going to help every every one of us in terms of quality, because it's going to improve developer experience. You know, it ended up that people were just copying a copy pasting code from there, you know, but even though it was not meant to be for production, you know, but so I think that that has helped a lot and I think we have a lot of stuff that we have developed has been like this based on either problems that we face, but this is my personal that I did.

Researcher 39:57
Yeah, so how it did help quality You said it's improved, or it helps developer experience because your, your, your team members, other developers start reusing it using the same code.

Participant 14 40:11
Yeah, and some of this is also in code keratoses, some of them are libraries, you can just import them, then it added a central place and instead of people replicating, you know, because if I told you, you have to educate yourself in such a way, everyone's would have been implemented, and then we ended up with 20 different applications. And then you know, it's going to evolve and they will become completely separate piece. Making the library beforehand ensured that it's gonna be at one central place. And this is what happens now it is a bug, I fix it there, and I just make any sound announcement, IT team channels, please just bump the version in your library. So they just have to change one line now. So yeah, that's,

Researcher 40:56
That's a good example. Thank you. And can you tell me from your own perspective, what encouraged you what motivate you to take initiative in this team?

Participant 14 41:07
For, for me, I, I really enjoy development, I like engineering in general. So and it was a technology was the I was running a lot learning is about really helps me, if I'm learning something new, I just have a lot of energy, you know? And I'm like, Oh, these are these, what can I do with this, you know, this is something interesting, and I read something, I need to just put this somewhere in my code just to test how it works, you know, so I have this energy that drives me. And also, I kind of think that from by making these, like, for example, a previous example making these sample applications, you know, and I made a server, but it also helped me look from the client perspective, the people who are going to be using this, how will this give me give me both perspective. And also, when I have both perspective, this is the me that made something and this is the media's gonna use something. By doing these things, I understand. I understand the pain points that they will be facing, you know. So when once I once I'm

working with this, me, I kind of like, this is not good, I need to improve the experience for myself. So that kinds of, but in general learning is something I really enjoy always learning new stuff. And that always keeps me going.

Researcher 42:27
So do you think the culture you have in this team entourage is your qualities as an individual, it's, it's, it allows you to learn, it allows you to create?

Participant 14 42:41
Yeah, I mean, it has always led him in this project was Typescript and nor, I joined this assignment is that this the first TypeScript project I created in my life was this project, you know, so it, it allowed me to learn all these things, you know, I had no experience with and I probably didn't even do an overtime or anything during this time when I made a full fledged application, a critical component just by running an entire programming language from scratch. So it was everything.

Researcher 43:12
Okay, fantastic. Thanks, Participant 14, I move to the next one, which about helping each other in the team? And we ask, it is difficult to ask other members to help and you said no, not at all. With my current teammate, I have been working for three years. And we can straight away or write to each other on team and help if someone is busy, they also feel free to say that they are working on something as at the moment, and they can help later. So you talk to me about pair programming, you mentioned pair programming earlier right? What do you use paper programming for to help each other?

Participant 14 44:00
Yeah, for pair programming is for help each other we have people from different experiences, some are, you know, they're straight from university so they don't know much. So sometimes it's very mundane tasks. So it's like you know, how do you do it so let's case share your screen and I tell you click here and do this. That's another way the other is a lot of pair programming is you know where we have a new story or new application to develop you know, and then we want that not only one person is should be just involved in it is good that two people are working on it. So if you have sickness are we to spread the knowledge, you know? So then I've been working with Gaia before meeting we were doing this peer programming, because it's a new thing. We want to spread the knowledge so I'm coding and the other guy is also looking at VR also like sharing okay, but about we do this, you know, sharing knowledge and so one person codes and the, you know, we're talking and coding is like this So it helps us to learn something new, if we are developing as a new service, and we have to collaborate with other components that the teams have made. So it goes to have two people working on each other one person always is, is the one who's coding, we might switch lucky, maybe it's your turn, maybe it's my turn, you know, just to change the base. And there's sometimes it's always the you have a production issue, for example, you know, and then you can have not via programming, but you can have, I don't know, four or five people just looking at one person sharing the screen. Other people, I will try this data logs there to something like this, we also have it on our computer as well, and the blogging, and I'm telling you, okay, I checked this here, you know, it's more like, you know, if it were five people were in the same room and looking at something together. But when you do it online, then someone person

has to share the screen. And other people are doing their bit. But when there is one central person, we're putting all the information each you know, so it's

Researcher 46:04
Okay, that's great. Do you think that knowledge sharing and helping each other this way in your teams improve the quality of the software you're producing?

Participant 14 46:17
Yeah, yeah, I think, especially when you're also doing like, pull requests, and someone says, like, review my code, and okay, is, is one way to write comments on the code, Lucas and ggsN. But it's, it becomes also good that if you're sharing the screen, and I'm telling, you know, like, for a junior developer, like I'm telling them, like, yeah, don't write it like this, you know, do it like that, and, you know, videos explaining to them and instead of just writing to them, then it's in a much better, you know, because they're guiding them through a process. And maybe it's like, Okay, I did it, once I told you, now it's your turn, try to figure out what I did just just repeat it, you know, so, and the way defects of equality, one is, that is spreading knowledge, that a person would learn something new, and then produce better quality. And, of course, when they are written code, by digit doing pair programming, you're kind of doing their code review, you know, so, and it really helps, you know, normally when one person is writing code, then the other person is reading them, and they're more acting like a linter, or a compiler, finding issues with them, you know, a second pair of eyes, it this always helps, you know, because I've seen myself, I write code and you know, writing code, making spelling mistakes, or you know, some writing something wrong. And if I had been by myself, then I would have run it twice. So why is not working? But what went wrong, you know, but so many times by colleagues, I know you wrote something wrong there, they should be like this, you know, fatally, it helps by building things faster.

Researcher 47:55
Okay, great, thank you, and move to the next one and wishes about it says no one on my team would deliberately act in a way that undermine my effort. And your answer says, Yes, we always appreciate each other help and publicly thanked them in team meeting. In my current project, the leader also appreciate the support and effort, especially when someone goes the extra miles. So this Antra team support? How does it make you feel about the team? And how makes the team work together? Because it looks like there is a lot of support?

Participant 14 48:41
Yeah, we have always been supporting each other. And unless I think unless we have like, Yeah, this is the priority I'm working on and I cannot do anything else. We are always supporting each other. So I wrote 10 times a day as we were writing each other, hey, I need help with you. You know, I asked my colleague today, I'm going to look into this, would you want to join me and gesture. And sometimes we do have issues and people get specialty in some, you know, different services. So they in the end date and turned out to be the one that's the most bugged if there's a problem happens there. So few last two weeks ago, we had a problem in a service and scrum master was reaching out to me, we have this bug and these guys in the factory in the client's office and stuff is doing stuff. So I'm looking a lot of stuff,

you know, and I'm doing my development as well and lots of stuff is going on. But then, then we have the retrospective and then disclaimer says thank you for your support. And I know that I put a lot of things on you, you know, it would be good if you can spread this knowledge but right now you were the choice, you know, so you were the best person to look at. So sorry about that. But thank you for the help. So there has never been any undermining anyone's efforts here.

Researcher 50:00
So do you have an example where somebody made an effort and next like you said in your answer, to achieve quality or to prove to improve the quality of the software?

Participant 14 50:15
Yeah, I mean, wonderful components add, add some, a, it was working, but it had some missing features that were not demanded or the stakeholders, so is also programming games, you know, even coding in his spare time. So he has spent, he says, Oh, I was working the entire weekend, I made this debugging will debugging feature is gonna help me and is going to help the team as well. And everyone was really happy, the stakeholders were happy, you know. So, of course, it was appreciated, many of the features in our application are like ideas. So, so many people have worked over the weekend to develop, develop things, you know, because you know, they don't have time for this right now, just a, we have to deliver this. So I have worked sometimes on weekends that I really want to get this thing done. And it's always appreciated our stakeholders, and they see they helped us even with support. Publicly, they write even volver they were sent mail, not to only the team, but also to our bosses and our parent company, as well, to know that, you know, there to let me to let my manager know that I'm going to be doing a good job at my assignment, you know, so I think it's written, it's really good that you know, that they always appreciate.

Researcher 51:37
How does it make you feel working in a team that always appreciate you?

Participant 14 51:42
Yeah, I mean, for me, it's it always motivates, motivates me, I work I like learning, and I'm all it gives me a lot of energy. But what what really puts it on steroids is when you have this manager, you have the team with the same motivation and motivation, and they support you on it. Because then you feel like, even then you work that you will be appreciated as you did something really good, you know, and, oh, you made a fancy feature, you know, but people like, you know, you really made a really cool feature, you know, even if we shared something like that, and, you know, it could be even become a like a healthy competition where you're trying to build this really cool feature to help the system. So I enjoy working is I think it's it's kind of a theme I tried.

Researcher 52:32
Yeah. Okay, thank you. There is an item which is similar, which is about appreciating your skills. I don't think we need to go through it, because it's similar. And you already answered that. So I will wrap up and with some two questions, overall, do you think this, this dynamic, and this quality of safety, helping each other, talking about mistakes not being blamed for mistake? Do you think it helps improving the quality of your software?

Participant 14 53:10
Yeah, I mean, I mean, definitely, I mean, there's the quality of the team and the relation among the people on the team, I think it will directly affect the quality of the software, you know, because we are in the NVR developers, where again, we are also human beings, our emotional state will affect the code that we write, you know, if you are unhappy, then the code that you write is poor quality. But if you're motivated, then you can see it also in the code, you know, that oh, this person has really spent some effort making, you know, he learned some really cool principles from YouTube's reading solid principles and stuff. And he's applied all those fancy, you know, JavaScript and the fancy language features, you know, and that was the proper motivation, you know. So I think it directly affects it. And maybe some companies might think that's one company I worked with was more like, detecting the bonus for if someone was not pretty had produced some bugs. And I think that was the completely wrong way of doing it. I think the migrant company, the proper way is they focus on the team being happy, or the people being happy, and that it indirectly affects the code as well. So if your software is just, I guess, just another project or another task, than I think that is completely the wrong way of doing it. We are happier because we feel safe and listened to and have a lot of freedom on our work.

Researcher 54:45
Okay. That's great. That's very, very interesting and deep statement. Participant 14, thank you very much. I'd like to give you an opportunity to, to add something I didn't cover In this topic, you know, we've been talking about the quality of safety in your team and things like that. Anything that I haven't mentioned, and you'd like to add?

Participant 14 55:14
I think it's terms of software quality. You weren't interested in, in the processes themselves?

Researcher 55:24
Yes. I, because you give me some good example. I mean, if you have any example relative to the process, yes, I'd like to hear it. Because the process, of course, we did talk initially, about what processes do you use? But if you have examples that helped you to improve the process you're using to achieve quality? Yes, I'm happy to hear it. Yeah.

Participant 14 55:50
Yeah, I mean, right now, the technology stack you use, like, it should be the time, the round trip time to publish code. And to figure out errors in production or in live environment, it should be as low as possible. And that things, that helps a lot, you know, if I'm going to write code, and now get two hours later, I will have the results in the environment, then I'm probably going to lose focus and forget about the bug, you know, when I go on a coffee break, now we have it, we can get it in production in less than 30 minutes, or even it runs tests. So 20 minutes is running tests. So that is very important. So that's based on the technology stack, you know, if you're using Docker and stuff like that, you can achieve that. And we have this continuous integration, that really helps a lot. Because previously, I've worked in companies where it was a release goes every once in a month, you know, and you have to ask this team to deploy, and it was just so hectic that you really can't really focus on something, you know, you need this for the feedback loops would be much smaller. And this is continuous integration. And all this

latest DevOps tools really help you to achieve that. So I think this also has to be really productive. You know, if you're compiling an old C++ things where it takes 710 20 minutes to compile, then you're just kind of like, oh, I made a bug. I made a typo. Now I have to wait 20 more minutes, you know, that guy that kinds of demotivates? You again, as a person?

Researcher 57:28
Yeah. But do you think these type of initiatives and tools, do they improve only production productivity or also quality?

Participant 14 57:39
Yeah, I think the quality will be quality, it does improve things like Docker does improve the quality because it allows you to produce more reliable environments for your software testing, visuals integration, I even took the initiative of using Docker and today to develop integration tests, you know, because I don't like this mocking things, I ran the real database with real everything to test with real things, and then destroy it. So it was for every fresh test, it was a completely new environment. So they they're like, really ensuring software quality, because we're not testing with fake databases and fake functions, you know. So they're, they're really helping and this, this is directly helping. And then of course, the time does the shorter feedback loops also improves the quality by, by at least allowing you to do more in less amount of time, you know. And then we also use things like link things like for figuring out bugs in the code, that's, again, that's directly affecting software quality, because it's gonna figure out some code smells, anti patterns, you know, typos and these kinds of things. Sometimes you have exception handling code that, that runs very rarely. And they could have bought boxes donated by donated and then code smells, tools, yes, lint and sonar cube, like tools, they have to figure out these problems, you know. So all these to do with software quality.

Researcher 59:17
So people are comfortable using linters. Because they can be confronting for some people.

Participant 14 59:23
Yeah, I mean, they, you, of course, we have configurability how much you want to configure sometimes, you know, this is a common issue. It shouldn't be like this, especially, it happens with older projects. I think when you're starting new, and when you start put this can lead into things from the beginning, you know, you make a proper boilerplate, you know, then then these things work really good. But if you add it to an old project, and it certainly gives you 5000 errors, no, no one's gonna fix them, you know, then it's very irritating. So it's, I think it's early adoption really helps with with with old projects is very useful. Big problem trying to fix those things. And we have to enforce some things, you know, these, these are kind of rules is good to have them. Some people not like them. They're one guy didn't like TypeScript at all, you know, he was JavaScript, you know, and but we convert him to TypeScript after five months, because he was like, Oh, this is really good. Now, it helps me. Now they see the benefit of it, you know, to show them.

Researcher 1:00:23
Yeah, you mentioned a good things because linters are rules, right? And you agree on those rules as a team? How can you balance between enforcing rules in this environment where people are nice,

healthy to each other? And how do you balance between rules and maintaining this safety work environment?

Participant 14 1:00:47
I mean, the latest we these rules for uses, we're not like defining every single rule, they are templates, like you have the Google template and the Airbnb template for your front application. So it more happens, like over making a front end react application. Airbnb template is a standard, we put it there, you know, and there's a rule we have, you know, and sometimes there are exceptions, you know, okay, we make turn this from an error into a warning, you know, something like this. And then some people are the biggest conflict we have is the width of the line, you know, people with big screens really say we want 200 characters, you know, but I say we don't have one not everyone has 4k screens, so give it 200 You know, so there's the bigger I think the the rules this is on the construct that comes in a big people with bigger screens and bigger resolutions want bigger, longer lines.

Researcher 1:01:42
I never realised that that's quite interesting. Okay, Participant 14, I, it's good to late and thanks a lot. I appreciate the example and I enjoyed the conversation and the discussion. Thank you.

Participant 14 1:01:47
Very nice. Have a good night. Thank you.