Participant 7

SUMMARY KEYWORDS
colleague, project, team members, team, learning, understand, people, discuss, quality, mistakes, questions, helps, implement, agree, safe work environment, improve, software, members, behaviour, approach

SPEAKERS
Participant 7
Researcher

Researcher 00:13
Hi, Participant 7.

Researcher 00:25
Hello. How are you this afternoon?

Participant 7 00:32
Yeah, I hear you. It's evening already.

Researcher 00:35
Oh, yeah. Well, it's evening almost here. So it's five o'clock. And thanks for your time. And thanks for the opportunity to do the interview. I really appreciate. Thank you. I do have a lot of questions and things to go through. So we will jump right in. And I'd like you to start with a brief introduction of yourself. Just introduce yourself, mainly your education, your experience. Briefly, please. Thank you.

Participant 7 01:09
Okay, so from my end, I started development 15 years ago. So while I've been working mostly on dotnet, and Microsoft technologies, I've been involved in quite a few projects in various areas, such as well, finance, home automation. And all sorts of web development projects. In the past couple of years have been mostly working on Cloud project for a medical devices. And well, during my career, I have worked with various people and with various methodologies for software development.

Researcher 02:09
Okay, fantastic. Thank you. And I understand in your current teams, you work on Agile, right?

Participant 7 02:20
Yes, I worked on as far as CSV. And our latest project is also in Azure. I've been working on DevOps has one. So I've seen quite a few technologies during my years. And I'm quite grateful to have used most of them.

Researcher 02:48
Okay, fantastic. So what are you using Scrum?

Participant 7 02:53
Yeah, we are we are doing Scrum. And we are focused on implementing Agile methodology. But it's quite difficult sometimes to implement it. Because on the client, the timelines for the project. Sometimes priorities change, because the client needs things completed for a certain day. And because of that, these methodologies are not always respected. Yeah, that happens quite unfortunate.

Researcher 03:43
Yeah, it's common. Yeah, it's, it's well known. Can you describe a little bit how do you use Scrum? How did you implement it?

Participant 7 03:54
Well, we try to have daily stand ups, where we discuss the project status, what we did yesterday, what we did, what we are going to do today if we have any blockers, that's kind of topic and we also have a sprint planning session where we discuss the next development features. We have sprint retrospectives, where we discuss if we have things to improve, where we are lacking, what we did good and what we should continue doing. We also have certain Well, in some cases, we have clear programming sessions. But that is mostly to help coworkers to ramp up or if they do not have a clear overview of the architecture and the code base.

Researcher 05:05
So you use them to transfer knowledge, right?

Participant 7 05:09
Yes, exactly. And I have seen that this type of session helps a lot.

Researcher 05:17
Okay, great. Thank you. Just a few question regarding the team. And we will move to the core answer to the questions, the one I've sent you on the email. So, in this team, how long have you been working together?

Participant 7 05:33
Well, on this particular project, with work, we work together for the past two and a half years. And while the team changed a bit, over time, had a couple of colleagues that remained from the beginning. But other colleagues, for example, we had few junior members. They tried to get on track as fast as possible. But even though they managed to understand the project, and what's it about, they still had some lack of knowledge in regards to how things were connected, how the services communicated

between each other. But well, as I mentioned, pair programming help. In some, in some cases, we had team members that proposed some improvements on the project, which were welcome. But in some other cases, those were not. So what those were not desired, because of the time constraints and the client priorities.

Researcher 07:09
Okay, thank you. Another question. How big is the team? How many team members are there?

Participant 7 07:17
Well, we had from three members to six members on the team,

Researcher 07:25
Why the number is fluctuating? Sorry, why the number is changing?

Participant 7 07:34
Well, the project went through several phases. And in the first phase, it was sort of like discovery phase, where we created certain POCs for the client to get the grasp of what's going on, and what direction he wants to go. And then we have certain presentations that needed to be made for Microsoft. And because of that, development was rushed to implement certain features available for live demo. And only after that we started, implemented, implementing thoroughly the platform and discussing all the small details. Then, the team got bigger.

Researcher 08:32
Okay, I understand. Yeah, that makes sense. So are you only developers, or you are cross functional and mixed of developers and QA or other skills?

Participant 7 08:46
Well, we have a few developers on the team. We have two QA members, project manager, a team lead from the senior developers, which is myself, and some developers on the client side.

Researcher 09:10
Okay, great. The last question regarding the team. What type of software do you develop? It is you said it's for medical equipment, right?

Participant 7 09:22
Yeah, it's software. It's actually a platform dedicated to analysis of it medical data. And it's a web application where I'm not sure if I can disclose more details about its functionality.

Researcher 09:42
No, that's ok. That's enough for me. I understand. I just wanted to know what type what type of development you do. So I have an understanding of what goes on in the team. I don't need to know more.

Participant 7 09:53
It's a web application. But it's has an architecture based on micro services? It's a combination of monolith application with micro service architecture.

Researcher 10:10
Okay, I understand. So, yeah, the web, the web part is just the front end it just the user, the client things. Yeah, that's fine. That's enough. So, we will be talking about quality in the example we will be discussing later. So as you know, quality can have slightly different interpretations from team to team. So to make sure that we understand the same thing, I will propose a definition we use and we can discuss it briefly and we can move to the to the core questions, right. So we use the ISO standard definition. And I will read it to you and you can give me your feedback on the definition. So the definition states, the ISO definition states, software quality is the degree to which the system satisfies the stated and implied needs of the various stakeholder and thus provide value. So this same ISO model proposed some non functional characteristic that the software should adhere to. And some of these are performance, compatibility, usability, reliability, security, maintainability, and portability. I'm sure you know, all of these software qualities. So to me what I need to understand whether you agree with this definition, so we can work based on the definition or you disagree.

Participant 7 11:42
No, that's quite accurate definition of quality.

Researcher 11:47
Yeah. Okay, great. We can move on. So what do you have as processes in the team to assure quality, what type of testing, code review, etc, you have in place?

Participant 7 12:01
there are quite a few things that we well, we should be doing and some things that we are doing. There's a quite a difference in project development in software development, in regards to what we should do and what we are doing. So, we should have theoretically, unit tests and integration tests and various other methods of testing. But those are sometimes implemented, sometimes not, depending on timelines, and also priorities. And we also need to account for deliveries for certain releases on certain scheduled. So unit testing is something that we are doing already, we are not doing integration testing so much. We are also doing code reviews for the pull requests that we make. The code reviews are quite thorough. Each member of the team can participate and include his or her observations. And we also have certain tools to help us out in identifying breaking changes in the code or reliability maintainability degradation of the code. We are using external tools, tools for that.

Researcher 13:46
So are those static analyzer or what type of tools?

Participant 7 13:51
Yeah, mostly static analyzers. Like we have one for security.

Researcher 14:04
Yes. Yeah.

Participant 7 14:06
Then we have another for the maintainability reliability and code smells. We use sonar cube. Which works pretty well. And offers also comments directly on our pull requests.

Researcher 14:25
Well, that's quite advanced, quality assurance practices you have in place so that's, that's not common, actually. So that's, that's really well, so next question is dose quality assurance practices. How about software engineering best practices? It's good because it stands for continuous integration, etc. Those all could be considered software engineering best practices. Do you follow some of those?

Participant 7 14:56
Well for the continuous integration We have pipelines dedicated to building the project and also releasing it to our DEV environment for testing. For development testing, we also have some coding standards in place.

Participant 7 15:21
We have some coding guidelines and formatting guidelines for the code base. We have some linters on the front end. And I think that's, that's mostly it. We have a lot of documentation on the software side in regards to SDLC. And how we should approach certain scenarios.

Researcher 15:55
Okay, fantastic. That's quite good. Well, let's get to the fun part of the interview. Thanks for sending those questions. Those answers on the email I appreciate so we're gonna go through them. Before we do that. My assessment from the questions from the answers you've sent is that you work in a work environment that we call very safe work environment, I'll explain what we mean by that. What we mean by safe is not security safe, but it's other. It's a work environment where it's provide a sense of security from repercussions. So what do we mean by that we mean that every team member feel it is safe to admit mistake, it feels that it's okay to propose initiative. They feel it's okay to discuss problem. And there is a sense of confidence that the team will not embarrass or reject or punish someone for speaking up. So there is a confidence and mutual respect in the team. And trust among its members, people can come forward can speak up and can take initiative, etc. So do you agree with my assessment that the work environment is highly safe as we define it here?

Participant 7 17:23
Yes, that is a fair assessment. Because each team member, each of the team members can participate with ideas and improvements. Mostly, if there are well, larger improvements that need to be made, then we create items, stories, improvement tasks, for that. Otherwise, we can just sign off on those changes and allow the colleague to perform those changes. There are what some of our values and the company I work for, are being courageous, fearless. That does not mean that we don't see the consequences, it just means that we are allowed to step out of our comfort zone. And that way we can Well, ideally improve on our code base and the quality of the software.

Researcher 18:34
That's, that's very interesting set of value. I'll get back to that. That's a very interesting one. Thanks for sharing that. But for this assessment of safety, do you strongly agree, disagree? Neutral agree or strongly agree that is highly safe work environment?

Participant 7 18:56
Yeah, I would say that I strongly agree with it. I have not seen any situation where something is off. So I strongly agree with that it's a safe environment.

Researcher 19:11
Okay, fantastic. I'm happy to hear that. So you said that the values promoted by the company or the organisation is being fearless which has created this environment of safety. How do these values promoted in the company?

Participant 7 19:34
Well, that that is one of the core values. Let me just see if I can pull off the whole list because I can't remember all of them.

Researcher 19:49
Yeah, they are difficult to remember most of the time. Yeah, I worked with some organisations like that, and they are difficult to remember.

Participant 7 19:58
Yeah, but we were really have five but right now I can't remember exactly sure which ones are. Yeah, so I get back to you on that one. But as far as I know, there were five values from which one of them was being fearless. And each one, each project would have to choose one of the core values as its main core value. And because of that, while it is encouraged certain teams to step out of their comfort zone, and yeah, it is quite useful for progress and the feeling of safety.

Researcher 20:51
Interesting. I like that. Let's move to the question, I've sent an email and go through them one by one. So, the first one said, if you make mistakes on your team, it is often held against you. You said now necessarily, we try to understand the context in the light of the changes made by the team. And also personnel changes, we also try to learn from our mistakes, and how to prevent them in the future. That's very interesting. Can you elaborate a little bit more before we work on an example?

Participant 7 21:34
Well, yes, so basically being safe environment, as you mentioned, even if we have certain personal issues or personal challenges in our day to day life, we try to be understanding with our colleague, maybe he forgot about it. He had some problems that they so we try to be kind and understanding. But we also point out what can be done better and how it can be done better. For example, a team member admitted making changes related to something that was not part of the feature he was working on. We told him that it's better to respect the agile practices and not make any changes that are not tracked

through JIRA or Azure DevOps. And another example would be a colleague that introduced some bugs breaking existing functionality in regards to a UI related feature. And that was mostly because he has one who has rushed to do those changes without testing thoroughly. But he learned from that mistake and improved his coding skills.

Researcher 23:15
Yeah, that's fantastic. Do you have a concrete example we can discuss together which is related to quality? For example, you mentioned one of colleague rushed a change to production without proper testing. Do you have an example like that?

Participant 7 23:35
To be honest, I don't have something that pops out in my mind. Something really, really bad. That happens.

Researcher 23:45
It doesn't have to be really bad. It's, it could be a mistake regarding quality doesn't have to be really bad.

Participant 7 23:55
Yeah, so the changes that our colleague introduced, were related to a page that contain the survey, which needed some medical answers, and those were related to changing the units that were used instead of milligrammes or the animals or something like that.

Participant 7 24:30
But we were quite understanding, because we knew that that part of the system was quite difficult and complex, and we had tones of questions with various medical terms, which we were not used with. And we helped the colleague solve the issue and then we redeployed. It was not production issue, but it was important for a certain level to be successful.

Researcher 25:07
Okay, fantastic. That's a good example. So he couldn't overcome the complexity. And he made the mistake which resulted on a defect that can happen, of course, in your answer, you said we learn from our mistakes. How did you learn as a team from that mistake? And how your colleague learned from that mistake?

Participant 7 25:35
Well, first of all, when the defect was discovered, we had the core line where I think we had a personal meeting, where we discussed and then we had a programming session to go through the fix and the changes and explain some of the complexity of the feature he was developing. And he also learned that being fast, or coding faster does not help, you run into defects. And it's better to take some time to think things over and analyse what changes were made, and then push the changes.

Researcher 26:30
That's excellent. That's excellent and healthy environment. Very healthy environment. So do you think his behaviour has changed? So did he adopt those learning in the future?

Participant 7 26:46
Well, for the most part, yes. But he also had some other small mistakes, but nothing out of the ordinary. I consider that as well, a learning situation. And whilst also I think he was actually looking for a different role, and the company. And that's also probably part of the reason that he may made some mistakes.

Researcher 27:29
Okay, but do you think that you saw some improvement overall, in his attitude to work this type of problems?

Participant 7 27:41
Well, his attitude, you mean, the way he perceived it or?

Researcher 27:51
So the behaviour was corrected. He stopped rushing and developing very fast.

Participant 7 27:58
Yes, yes. He tried to take his time to improve, as we all know, we are not perfect. And we make mistakes again, sometimes. Because of various factors.

Researcher 28:14
Yes, yeah.

Participant 7 28:17
He tried his best after that.

Researcher 28:19
Okay. So it was a positive and constructive experience, which helped him to improve. Would you agree?

Participant 7 28:28
Yes, I strongly agree with that. Yeah.

Researcher 28:30
So do you think it's affected the quality of his work going forward?

Participant 7 28:37
Definitely. Yes. Because each mistake we run into if we have the proper guidance, and the mentor that can explain to us and well, in kind and calm and constructive manner, it will help us improve in the future. And at least think before we run into that mistake again. This did not apply to this colleague in my example. I observed this in every team.

Researcher 29:05
Yeah, that's a good thing to see. I mean, that's the minimum you would expect people to reflect before they try the same thing. So there is a change in the behaviour. People change their behaviours.

Participant 7 29:21
Yes, it's not. It's not that immediate, sometimes, but on the short medium term, you can see some improvement.

Researcher 29:35
Yes, it takes time to change those type of behaviour, but you see a little bit of improvement and sometimes the person may make mistakes again and learn again, but at least like you said, you see some improvement taking place, right?

Participant 7 29:52
Yes, yes, there is some improvement and we also need to understand that mistakes are part of the learning Path. Yes. If we don't make any mistakes, then we would not learn anything.

Researcher 30:10
Yeah, something is going wrong, right?

Participant 7 30:14
Elon Musk said that as well.

Researcher 30:16
Yeah. Thank you. Thanks for the good example, we will move to the next items, which was part of the email. It says member of your team can bring up problem and tough issues. And you answered yes, definitely. It depends, though, how we react to them. Tough issue should never be dismissed until they are clarify within the team. That's that shows a very healthy work environment. Right. So can you can you elaborate a little bit on this?

Participant 7 30:50
Yeah, sure. So, in regards to problems and tough issues, there were some when some issues related to communication in certain regards, between colleagues. And it seemed that doing remote calls, between those colleagues did not help much. And usually that that is resolved by having face to face conversation. Or in in a group situation, where we get to know each other better try to understand each other what each team member is, is passionate about what his or her goals are. Because if we are just having calls over the internet meetings online, it does not help with the communication skills. We also need to do it face to face. Okay, that's on the communication side. But, yeah, we had some other issues

where we were a junior member was trying to improve the code quality on the front end side. And he wanted to add a code linter. But we talked him through and explained to him that those are big changes, because we needed to update a lot of the code to account for the linter. And that would have probably caused some other issues. The reason for adding that linter was the fact that developers might write the wrong syntax or something like that. I was a bit adamant about it. Yes, it can happen. But because the impact was quite high, we decided not to do it. Or, for example, there were some problems in regards to the way we were accessing the API from the front end. And while the way, one of the members approach this was by adding layers on top of layers to simply fetch a URL in JavaScript, so I was not that happy about that.

Researcher 33:47
So what did you do in that case? I mean, that's not a good design. It makes it less scalable. That's what I understand the right. Yeah. So how did you deal with it?

Participant 7 33:59
Well, it increases the code size and the build size and so on. But I had to intervene with another colleague to try and explain that. This is how we do things on the front end. We don't import design patterns from the backend on the front end. Just to implement Design Patterns. We need to understand each component of the system and certain components we have certain guidelines and other components, we have other guidelines.

Researcher 34:55
So how did you approach it and how did he receive it? This this constructive communication about this design decision?

Participant 7 35:05
Well, at first, he did not receive it so well, because he was adamant about his proposal. They had to, well, I had to call another colleague and try to explain to him together. Because he was, he was a bit stubborn on the issue. And he understood eventually, that it was, what his intentions were good. But the execution was, was not so good.

Researcher 35:43
So what do you think he made him accept at the end, because you approached him in a constructive way or because the environment is safe, or when the environment was safe?

Participant 7 35:55
I did not have initially the right tone with regards to the rejection of the changes. But after we talked together with our another colleague, in a calm manner, he understood it better.

Researcher 36:20
Okay, so the tone matters. Yeah, of course, the tones matter. And it says a lot of things. Yeah, the way you approach people the way the tone of your voice, of course, it sends a lot of signals. So you avoided a design issue, which could have make your, your part of your front and less maintainable and less

readable? So yeah, so you, you did that? Do you think his behaviour has changed after that? He learned something from this from this?

Participant 7 36:59
Yeah, I think so. Because he is a smart guy. But even if you are smart, sometimes you have certain ideas, and you get stubborn about them. And that's where a group discussion or some, someone that you like, more, that advises you on certain topics may help.

Researcher 37:29
And the problem with smart people, they, they don't get challenged much in their lives. It is difficult to tell them you are wrong, because most of the times they are not challenged much.

Participant 7 37:46
Yeah, so I would say the same thing. He probably did not find too much challenge in what he was doing, and he tried to make it a challenge.

Researcher 37:58
So another side of this constructive and healthy approach to this is you and your colleague approached him to correct the situation? Do you think you approached him? Because of the safety in your workplace, you felt it's safe? It's okay to do so.

Participant 7 38:20
Yes, definitely. As I mentioned, we have certain values in the company that allow us to be open about issues that we have to discuss with our leads with our project managers, people in charge. So that that has been from the beginning.

Researcher 38:43
Okay, fantastic. Thank you. That was a really good example. Let's move to the next one, which is about rejecting someone for being different. We could use the same example of your colleague who use the design patter of the backend, the front end, but I'd appreciate a new example. So the statement says, people on your team sometime rejects other for being different. You said no, never seen this before. Everyone keeps things professional. What we mean by rejection Here is somebody has a novel or an orthodox idea to approach something or he took an initiative to improve something. So in your team, when people bring new ideas or a different way of working, so do they get rejected? And now they don't get rejected?

Participant 7 39:45
Well, at least not immediately. People are first listen to we have certain conversations we point out the problem And the columns and also the constraints on the project. And based on that we agree on certain one certain outcome.

Researcher 40:14
Okay, do you have an example? Or somebody brought? Or suggested something? And which take it? Take us to the next question because they are related. So I'll move to the next one, which is quite related. So it is safe to take risk on your team. And you said mostly yours, but it depends on the context. If the client is breathing down your neck, I like that one. You should make the risk worth a while with benefit and outwait it. So what you say in here, you always assess the risk and the initiative, whether it's worth taking it or not, right?

Participant 7 40:57
Yes, because you might actually have some risky changes, that might improve the performance of the code base by 20 30%. For example, we had a change for going from dotnet, three, one to dotnet, five, and then dotnet, six, and we had some risks there because we needed to update the code base on a lot of components. But the benefits were ability to host the project on Linux containers, and that also reduce the costs and improve the performance. And also the fact that we move from version to another. That in itself, added some performance benefits.

Researcher 42:03
So this initiative, which has improved the performance, you as a team, you decide to take that initiative.

Participant 7 42:15
Yes, mostly as part of the one the investigation tasks and learning that we do from day to day, being up to date with the technology news and performance improvements, and certain ways of approaching problems. That helps.

Researcher 42:44
So, you said some very interesting things, everyday learning, keeping update to the technology. And in this example, you shared with me you took an initiative. So do you think you do so you take initiative you learn and you keep update to technology? You think you do that? Because of this safe work environment that you have?

Participant 7 43:14
Yes, definitely. That is one of the factors. The fact that the company allows you to grow and take some changes, chances helps a lot. It allows you to experiment, certain ideas, certain technologies, and if those technologies are a good candidate than they are proposed, and possibly implemented.

Researcher 43:48
Okay, fantastic. So you shared with me an example of upgrading the dotnet framework from one version to another version, and that helps performance. Do you think in general, this attitude of encouraging the team to take initiative, does it help also the quality of the software you're producing?

Participant 7 44:16
Well, it helps on the non-functional and functional sides. But I think more the non-functional side. But there are also other factors that need to come into play. For example, being aware of how the agile methodology works, how Scrum works.

Participant 7 44:37
Trying to always ask yourself, What am I doing now? What Why am I doing this?

Participant 7 44:46
Trying to get answers from yourself when you do things. That also helps so it's not just okay, I found out about this. All technology and I'm going to implement it. It always comes to the question why? Why should I do this? Is it because of me? I'm eager to learn this. Am I being? Am I being selfish? Or is it because I really see some value in using that technology?

Researcher 45:25
So you weigh the you weigh the decision, it's not a life decision. Right? So you take in consideration a lot of factor and you wait, you make a wise decision. That's what you say in right.

Participant 7 45:41
Yes, yes.

Researcher 45:43
So how do you come to that wise decision? Is it but just by yourself or by consulting your team members?

Participant 7 45:52
Well, that is usually part of is done by consulting with key members. It's not done in isolation by me or because I say so. But sometimes, it is because I say so, I have be the middleman. Being a team leader allows you to enforce certain changes. But I'm also while I allow the team members to add to the quality of the project. But obviously, we need to discuss those changes in a group setting.

Researcher 46:42
That reduces the risk, right? When you collaborate together, it reduces the risk instantly,

Participant 7 46:48
Because it also it's also a matter of experience. Someone that is more experienced and has gone through more projects and has seen a lot of mistakes. Learned from that, while someone who is younger, might rush things might try to prove himself. But he or she needs to understand that you cannot evolve rapidly. There are some steps you need to take.

Researcher 47:24
Yeah, of course. So the senior brain, that's where wisdom they bring that balance, they slow down things. Yeah. So but still it happens in a healthy, constructive manners.

Participant 7 47:28
Exactly. Yes. Yeah.

Researcher 47:30
Fantastic. Very nice to hear. So I move to the next one, which is the most important in a software development environment, which has to do with helping each other. So it says it is difficult to ask other member of your team to help you said no, never. The only problem is finding time slot to discuss in depth about problems. Sometimes that makes sense. As long as you approach people in in a constructive and productive way. You ask them for time and things you willing to help each other, right?

Participant 7 48:16
Yes, yes, definitely. Mostly, it's a calendar issue. It's not? Well, I don't want to talk about this problem, or I don't like this person, or it's time based. Sometimes there are a lot of meetings, which take time. And because of that, when certain questions are not answered, face to face, or in a call, but via email or text.

Researcher 48:52
Yes. Sometimes it doesn't have you need to be face to face or like with the wind. Yeah, I agree. It changed. It changed the dynamic. I'd like an example where you or a team member help another team member and it has helped some aspect of the product you developing quality.

Participant 7 49:19
So could you repeat the question?

Researcher 49:22
Yeah. So you help each other you transfer knowledge using pair programming, right? Yeah. So do you have an example where you or a team member helped each other to improve the quality of the code the quality of the product, so you help each other in order to achieve a better quality?

Participant 7 49:46
Yeah, definitely. For example, we had some feature to develop in regards to exporting files to excel and CSV. And when the junior member had some issues with that, because we have a pair programming session, it was done online. But nevertheless, it helped guide him through the why certain things are how they are implemented, where certain things are understanding how to better investigate certain issues when developing. And also expanded on that. And we expanded the discussion to other components in the architecture, which help provide some answers to questions that he did not get to ask.

Researcher 50:54
So, you helped him to learn in that programming session. So, do you think the quality of his work has improved because of that learning?

Participant 7 51:06
I would always say yes. Because every learning opportunity that you get helps you, not by 100 fold, but every small step matters. Eventually, quality improves.

Researcher 51:20
It's build up, it helps. Yeah, it's a continuous process. Yeah, it's a continuous process is not. It doesn't happen in one day, right? Definitely. Yeah. But from your experience in this safe work environment, learning from each other, helping each other overall, does it help to produce better quality? Do you see that the quality is improving, because people learning all the time, people are helping each other all the time?

Participant 7 51:52
Yeah, that definitely it helps. But another aspect of it is the way management approaches this. Because if you take people, team members out of the project, and move them to another project, and then another team member comes in, and then you need to start again, with the knowledge sessions, that does not help with the code quality. But if a member remains on the team, it definitely helps even better.

Researcher 52:22
Because when somebody leaves, the knowledge goes away. And when somebody comes, you have to bring them up to the standards of your teams, how do you like to do things? What's your expectation from quality? Right?

Participant 7 52:37
Yeah, and that's very time consuming.

Researcher 52:41
Yes, it's very time consuming, and the person will need time to adjust and understand those expectations as well.

Participant 7 52:50
Yeah, and from what we've seen, it takes anywhere from three weeks to a month or a month or two, to get to know the system and will be comfortable going about.

Researcher 53:09
That's a short period for this type of system you develop because they are quite complex.

Participant 7 53:15
Yeah, but when, when, when we did the knowledge sessions, though, those knowledge sessions and pair programming are quite long. So sometimes we have like two hours of or, yeah, of pair programming helps.

Researcher 53:33
Yeah. So it takes your time away. So you I'm not productive. You're sharing knowledge. So you're not productive? Yeah.

Participant 7 53:41

Well, that's that's a separate discussion, because as I learned from I'm not sure if it was a book or not. The fact that we are doing pair programming does not mean that we spend only one hour of his time productively and zero hours of my time on productively. It's like 1.9 hours productively.

Researcher 54:10
So it's a win win situation.

Participant 7 54:13
Yeah, yeah, everybody wins on that front.

Researcher 54:16
Everybody wins. And in a safe environment like this, it's a win win situation. You're learning he is learning. He will be productive at one stage.

Participant 7 54:25
In my conversational my communication skills, my soft skills. My also my understanding of what I know on the project, and also his questions helped me out to identify maybe possible issues.

Researcher 54:45
And questioning the way you do things as well.

Participant 7 54:48
Yeah. Even questioning myself is sometime is very beneficial.

Researcher 54:53
Yes, of course. So when new team members comes because this type of environment you need openness. I think we discussed this earlier. When a new team member comes to your to your team, do you think it's easy for them to become open to share to admit mistakes, or it takes also a while before the person get this this, this this mentality or this mindset of exercising these values?

Participant 7 55:27
I think it depends. One, depends if the person was part of the company before. Second, it depends on his personal openness as, as a human being. If he is open, he will probably not have issues ramping up and communicating with the other team members. But otherwise, he might have a period of adjusting with the team members understanding each of their roles and their responsibilities and how to approach them.

Researcher 56:05
That's an excellent answer. I agree with you 100%. Do you think these work environment the more safe they are the easier for the person to become open and adjust with these type of values?

Participant 7 56:21

Well, and what I've seen till now, I would say yes, it helps. But there might be people that see this kind of openness as something bad, something negative. And they might close up because of that. I agree. I did not encounter this, but I'm just telling you my opinion.

Researcher 56:49
Yeah, of course. It's not for everybody. I agree with you. Yes. To be open to admit mistakes to admit, constructive criticism. There is a lot of vulnerability on it. It's not easy for everybody. Yes, I agree with you. There are people who wouldn't like to work in an environment like this, of course. Yeah.

Participant 7 57:13
Yeah, it's quite difficult for some people to express themselves in, in an open environment. Yeah. depends on the company values also.

Researcher 57:26
Okay. Thanks a lot. Participant 7, I really appreciate. There are two other items that were in the email. But we already discussed them. And we already covered them in your example. We can do the last one, for example, working with my member team, my skills and talents are valued. You said yes. My skills are valued use also outside my current project. So when you add values, how does it make you feel?

Participant 7 58:01
Well, it makes you feel appreciated. You see that your work, has value has impact has meaning. It helps, at least if not on your current project. Maybe some other colleague has an issue that is solved via something that we have on our current project, some technology, and that is quite satisfying. releases dopamine.

Researcher 58:32
Yes, it does. Do you think that when it's the same thing apply to quality when people work, or code quality is appreciated and valued? Do you think that they keep producing quality work?

Participant 7 58:51
Well, I think that there's a feedback loop there. And if you provide quality for your product, then the team gets recognition and is applauded. And that in turn would make them improve the code quality even more. But some people as you may know, also get lazy if they are. They are applauded and recognised.

Researcher 59:30
Yeah, it can have the other effect as well. Yeah, yeah. Yeah, of course. We come to and thank you very much for sharing those good example. Do you would you like to add anything before we conclude? Anything that we haven't covered or talk about?

Participant 7 59:52
I don't think so. I think we covered a lot of ground. So I’m happy. I think we covered everything.

Researcher 1:00:07
Yeah, I'm very happy. Thanks for sharing those good example and thanks a lot. I appreciate and I wish you a good night and we will stay in touch I'll get in touch. Thank you. Bye Good evening.