Participant 13

SUMMARY KEYWORDS
software, problem, team, sprint, people, ideas, quality, bugs, qa engineers, safe work environment, questions, third party applications, colleague, helps, environment, improve, implemented, manager, initiative, work

SPEAKERS
Participant 13
Researcher

Researcher 00:14
Good morning Hello, good morning.

Participant 13 00:31
Morning. I just went to grab a chair. Sorry.

Researcher 00:33
Okay. Take your time. How are you this morning?

Participant 13 00:37
And good. Thank you.

Researcher 00:39
Okay. Let's start because I have a lot of questions and things to go through. So before I do that, thanks a lot for doing the interview I appreciate specially a Sunday. Thank you. Thank you. So, perhaps we can start with an introduction, but the interview, I do have questions and I will go through my questions. And I will use the example you have sent me the answer you have sent me via email. Thanks for that and we should be able to finish in one hour. Okay. Okay, perfect. Okay, perhaps we can start with a brief introduction, just your education and your experience in software engineering.

Participant 13 01:27
Okay. I originally graduated as an electrical engineer, and then started my career as a software developer, or a software engineer. And then I it's been a bit more than six years now, since I started. And I worked for two companies so far. Both based in Germany. And I along the way, I also worked for startup project for almost a year. About that was more like a side job for me. About one or one or two years ago. And, like, language wise, I always worked with Java, working as a backend engineer. Maybe your questions can open up and we can talk about more in details.

Researcher 02:30

Yes, yes, we will open up a lot in the upcoming questions. I'm just curious to know, how did you become from an electrical engineer to a software engineer? How did you learn software engineering and programming?

Participant 13 02:44
Okay. First of all, since my childhood, it was actually my dream to become an electrical engineer. Okay. And when I actually got in the university, and started taking classes, I thought, maybe that's not what I was thinking before I got there, because sometimes, maybe you have a hobby, but when you start taking lessons about it, it doesn't become as fun as you thought. Yeah, that might be the reason but I had, I think, three classes a week regarding programming like C++ and a few others as well. And then those were the those were my favourite classes of all the other technical related classes. Therefore, I almost wanted to expand my career on software development, because I thought I would enjoy it more. That's, that's how it actually started. I had bootcamp in programming, I had my other senior friends teaching me. And that's how I learned.

Researcher 03:56
Okay. Just curious when you have a good American accent where it comes from?

Participant 13 04:03
Well, I don't know if I should say thank you, or

Researcher 04:07
It's a compliment.

Participant 13 04:12
Thank you. Well, yeah, for sure. I think when it was back in 2015, when I went to the USA for the first time, and I stayed there for twelve months, and then and then I thought to myself, then maybe my English is not as good as I thought, so that I should improve a bit about it. And then and then after that, I focused on a bit how I can speak with an accent or without an accent, actually. And then I took also some classes on this and watched a lot of American movies and stuff

Researcher 05:00
Doing very well. Thank you. Okay. Let's move to some questions about your team. Currently you using Agile. Right?

Participant 13 05:09
Right Scrum.

Researcher 05:10
Okay, fantastic. So how big is the team?

Participant 13 05:15

It is currently, we were about 20 people. And we and then after that, they decided that they should divide the teams into two because we were getting too much. And then right now, we are 13 people. We also hired more people on the other team. So both teams are now 13 people each. A bit a bit more than the half is developers. And then the rest is QA engineers, product owners, Scrum Master and so on.

Researcher 05:47
So you are cross functional team. Right.

Participant 13 05:50
Right. Right, exactly.

Researcher 05:53
How long have you been working together?

Participant 13 05:56
It's been eight months.

Researcher 06:01
Okay, so it's a newly formed team. Right?

Participant 13 06:04
Right. Also an international team, half the team is based in Germany, and the other half is based in Turkey. Okay. We work remotely all together.

Researcher 06:17
Okay, so what type of software do you develop?

Participant 13 06:21
We have a [deleted to preserve the researcher/participant anonymity] the company's [deleted to preserve the researcher/participant anonymity] And the website allows [deleted to preserve the researcher/participant anonymity] customers to insure their own cars. So it's basically an Insurance application for buyers or [deleted to preserve the researcher/participant anonymity] dealers, to insure the cars without going to insurance companies.

Researcher 06:48
Yeah, so this is a new product, or you're enhancing existing functionality, or you're building it from scratch.

Participant 13 06:57
There was an existing application to do this job in in one or two companies. But the product that I'm working on is actually a new one. It hasn't even gone live yet. I think they're, they're planning to go live in three months from now. And then once this goes live, it will take down the other existing ones. And all the country, it'll spread through all the countries over time.

Researcher 07:30
So it's a replacement of an existing system, right? Yeah, exactly. Yeah. Okay, fantastic. Okay. We will move to the most important question. So the focus of the entry view is I've sent you some question via email, but the focus will narrow the focus around software quality, okay. But we need to define it, you know, a software quality can have different definitions. So I will, I will tell you the definition we use, we use ISO standard definition, and I'll read it to you and we can discuss it briefly afterward. Okay. So, the ISO definition states that software quality is the degree to which the system satisfy the stated and implied needs of its various stakeholders, and thus provide value. The ISO model also proposed and cover some non functional requirements are characteristic of the software, namely performance, compatibility, usability, reliability, security, maintainability, and portability. So first, do you agree or disagree with this definition? Or would you like to comment on it?

Participant 13 08:52
I think I think that's, that's pretty much accurate to me, especially in terms of performance. That's one of the most important indications to software quality.

Researcher 09:08
Yeah, to you as a team, it is more important, right? Or do you think in general, it is more important.

Participant 13 09:15
I think, in general?

Researcher 09:17
If so, what do you do to assure quality in your team? What type of practices and processes do you do to assure quality?

Participant 13 09:31
Yeah, we design hack days, from time to time, and all the team gets together for two days. And then we also hire a professional hacker to help us along the way and then we try to hack our system to find some vulnerabilities in terms of especially security of the code. And then the next thing we do is we have also periodical refactoring sessions, where we focus on the code and see if there are parts that needs or that could use some help of refactoring. And then and then if we think so then we allocate some time in the next sprints, to do the refactoring. And we also are working on the unit test coverage. It's not the best right now. But we are all aware that we should focus more on that. So with each sprint, we are focusing more on writing more unit tests, actually.

Researcher 10:38
Okay, and who does the end user testing? Do you have a QA team?

Participant 13 10:46
We have two QA engineers in our team. And they do the tests that the end customers want to be done.

Researcher 10:59
Okay, great. Thanks for that. And I'll, I'll go through the question. You've sent via email, and I will tell you, what's my assessment of the workplace and get your feedback and comment. So we think that the workplace where you work at the moment, it's highly safe work environment, I'll explain what do we mean by safe? So a safe work environment is an environment where everybody believes that admitting mistakes is okay. People admit report mistakes they believe and if they feel it's acceptable to report and discuss difficult problem. The team welcome initiatives people take initiative, and they feel at ease at doing initiative and helping each other. So from the answer you're provided, I do believe that the team is a high safe work environment. Would you agree or disagree with this statement? Would you like to comment on it?

Participant 13 12:13
I definitely agree. And I can say that, we have this product manager that we see in every sprint review, sprint review. And he always says that, He always encourages the team that he says that the team is doing a great job. Let's keep doing this. And he always says, If you have any problem, don't hesitate to come to us come to me directly. It doesn't matter. And I will do whatever is best for you. And inside the team, we have individual, let's say a bit more problematic persons. Not that big deals, but individually. There are a few. But overall, the team and the concept is very, they welcome if you have any problem. They want to listen to you. And they definitely value your decisions and input. I can say that.

Researcher 13:14
Okay. Do you think that support that you get from the management, it's the this is what's made the work environment with this quality, a safe work environment?

Participant 13 13:28
I'm sure they value this approach this safe approach. And what was the question again?

Researcher 13:49
So you said that's a good answer. Yeah. So you said they value this safe approach to work environment? Do you think it helps to sustain and maintain this safety in the work environment?

Participant 13 14:04
Oh, yeah, for sure. Because I think these kinds of approaches start from top to bottom. So higher hierarchy people, if they're welcoming than the, the lower degrees are also affected by it.

Researcher 14:21
Okay. Okay. That's good to hear. Thank you. So I think we will start with discussing the answer you provided. So the first question was if you make mistakes in your team, it is often held against you. And your answer says now the team usually thinks it was mainly because of lack of communication time or restriction or poor planning. So people, when they do mistakes, my understanding, it's okay there is no repercussion. There is no blame is done. What you mean here?

Participant 13 15:02
Right? Personally, when I make a mistake, I blame myself for it, and I feel bad about it. But the team doesn't hold it against me. Or they usually say, okay, we should just plan it with better estimation in the next sprint, it was our bad that we couldn't foresee the difficulties. Therefore, I don't feel blamed or under pressure when I make a mistake.

Researcher 15:29
So why do you blame yourself?

Participant 13 15:33
I think maybe it's a bit about my personality that I always want. I'm a bit perfectionist, in my opinion, and I always want to do better. And when I don't do sufficient, maybe I blame myself. That's why

Researcher 15:52
Yeah, I understand. Do you have an example where yourself or another team member made a mistake, and it's affected the quality of the code? Or the quality of the product? Or it's broke something in production? Or? Yeah, so it's quality related somebody yourself or somebody else has made a mistake, and he or she was not blamed, and how the team dealt with it.

Participant 13 16:28
Currently, when I when I think about it, I honestly don't remember such an example. Because we, we talk as a team, we talk to each other very often. And if there is a problem, then we catch it before it gets bigger, it gets a bigger progress problem. Therefore, maybe that's one of the reasons we don't face such difficulties as often. It doesn't indicate that it won't happen, but so far, either I don't remember or maybe it hasn't happened.

Researcher 17:06
So when you discuss what do you can you give me an example you discuss about something and you prevented the problem from happening?

Participant 13 17:19
Let me think. For example, in in, at times we have, we faced problems, let's say our application becomes broken on our development environment. And, and then, or someone is having a hard time fixing what's broken there. And then this, this topic always brings up to our daily meetings or other gatherings whenever possible. And then the team focuses on it, if the person needs help, with support, they always provide help because the software shouldn’t be broken. And it's important to the team. So we, we always allocate people to help fix the bugs and things. I think, at one time, now I remember one, I think it was a the same day that we had a sprint review where our manager, our own client was there and be the screen was just broken, the application was just broken. And then we couldn't show much to them. But they just said it's okay, just like them. Tell us what happened. How, what are the things that you implemented? Just tell us tell us vocally and in the next sprint review, you can also show us it's not it's no problem. So that was an example.

Researcher 19:03
So that's quite serious in some places. I mean, that's a big mistake. I mean, the client came in for a demo, and the application was broken. So what has broke the application in the first place?

Participant 13 19:18
We our software also communicates third party applications. And by the time this happened, we didn't have a solution in case our software doesn't communicate to them in case let's say their software is down. And that was the case actually, the third party application was down for the whole day. And we were not able to communicate, but it's also our fault because we should have foreseen that this would happen and maybe mock the responses that are coming from them.

Researcher 19:55
Yeah, that's a serious bug. I mean, if you go in production without catering for that scenario, your side of things your application will creates a serious production issue and it's gonna stop providing the service. Right.

Participant 13 20:02
Right, exactly.

Researcher 20:05
So that's what I mean by a mistake. And you said you were not blamed for it right?

Participant 13 20:12
No, no.

Researcher 20:15
So the management had a very constructive approach to it from what I heard. Right. Right. So, so what, how did they approach it? And how did you feel about it?

Participant 13 20:34
First of all, I think there are people in our team that reports to managers within the sprint, as well, maybe every week, or I don't know how often they do it. But the managers are always aware of how things are going. And not only from Sprint Review, to sprint review, but they have more feedback. Therefore, I think they know that we are improving, or if we have an issue, and if they can help, they always go into place during the sprint. And then if there are necessary emails to be sent, they do the communication, so that they are actually in it somehow. But they don't really tell us what to do, or they don't put pressure on us during the sprints. And, like, at the end, we just realised that we didn't have a solution for the API communication situations with the other third party applications. So we, we didn't have a visual presentation for the sprint review, or the manager level. But we just told them the features, we implemented the bugs that were fixed. And then we also had a few things that we could actually present so that we kind of saved the sprint review, but we just had to present what was missing in the next sprint review.

Researcher 22:16
Okay, fantastic. So that was a very healthy way to approach the problem. And how did you feel about it as a team? And did you learn from it? And how did you learn from that mistake?

Participant 13 22:29
First of all, the day after this happened, we had our sprint planning. And we originally didn't plan a solution for this during our sprint refinements. But we thought it's worth to cover this problem, because this is serious. And if this happens, again, we can't explain anyone about this anymore. Therefore, we immediately added a story in our in our sprint, and then made the solution for this. And no one really talked about what happened that they or why it happened. But we just had a solution, and then it's forgotten.

Researcher 23:14
So you felt it's normal to That's my understanding, you feel it's normal. It's a mistake. You, you, you address it, and you prevent it from going to production. And you moved on. Right.

Participant 13 23:27
Right, right. Yeah, maybe that's why it took me a while to remember. Such a serious thing.

Researcher 23:35
Yeah. Because when you deal with mistakes in this healthy way, you don't see them, you don't memorise them as mistakes, because the place is so healthy, and nobody blames nobody, so you don't remember them. So it becomes a second nature.

Participant 13 23:37
Right. Right. Very true.

Researcher 23:40
Okay, thanks for that. That was a good example. Thank you. The second one, which is also important. And I'd like I like your answer. And it's about bringing problems and tough issues to the team. And the question asked member of your team can bring up problems and tough issues you said, Yes, we discuss any kind of problem and bring to the table during our daily meetings. Can you give me an example of somebody, it could be yourself or a team member who decided to bring a problem to the team and to discuss it? And perhaps if the problem is related to your software quality processes or a quality issue would be even better? Thank you.

Participant 13 24:53
Okay, this, this happened before we separate it into two teams. So when we were just one team is this happened and we had this problem in our software, where going from one page to another one would take a lot of time. So it was not customer friendly, or user friendly. And actually, it was me that came up with an idea that we should solve this problem. Because this is, it's serious to me. And I also not only mentioned the problem, but I also brought a solution to it. Of course, I made my study before bring it up. So I was prepared to answer the questions that may ask. And of course, they asked a lot of questions

like, did you think about this? The Did you consider that this would happen if you say that, and, and then I answered all of their questions. And then they just accepted it. And then they, they also gave me a chance to do it in one or two sprints. And it went well. But there was also another time, where I also suggested that, that there's a problem, and we should fix it. But then they said that this would be over engineering. And it's not really worth to put the effort in there. And I also understand that not everything has to work as a best practice. Sometimes you just have to go by the flow. But maybe those are the two examples.

Researcher 26:33
Those are two good examples. Thanks for that. And I'll, I'll ask a follow up question in both of them. The first one, your initiative was accepted by the team, and the second one was not accepted. So let's work with the first one which was accepted. So what was the result, the performance and the usability of the page was improved.

Participant 13 27:00
The page was improved. And it's not only that, the reason, the transition from one page to the other one was because there was a lot of communication going on with third party applications or within our applications. And when there was a problem with any with between any one of them, it was difficult to track where the problem was, and my solution also fixed that too. And then we were able to find the where the problem matured more easily. So it was appreciated at the end of the sprint.

Researcher 27:39
Okay, great. How did you feel about it? And how it motivates you in the future to take initiatives like that?

Participant 13 27:49
Right, I definitely feel encouraged for doing this more often. Therefore, let's say if they didn't think that this would be a solution, or they didn't give me a chance, it would discourage me. And then I wouldn't look at the software with the perspective that like, is there anything I can improve in the software, but now this is my aspect is this is an approach or perspective I gained after this example. And now I look at the software always and think if there are things that can be improved. So if everyone thinks this way, and feels encouraged to do this, I think it's a helpful approach for the whole team to improve the software as a whole.

Researcher 28:37
And do you notice that other team members do the same they bring initiative forward?

Participant 13 28:45
Some of them they do. But they said before I joined the team, maybe a bit more than a year ago. They were not as welcoming to these kinds of ideas. And then because I think I briefly mentioned one of these in one of the other questions that you sent me in the email. And maybe I should just open it up a bit. So before, it was a year, almost more than a year ago, and when a very senior level colleague came up with a solution to a big problem. But that would also cause a big change in the base architecture. So it's a serious thing, but definitely sounds like it can bring a lot of improvement and

optimization in the software. And it was not welcomed from the architecture team. And no matter how many times he tried to explain it didn't go well. And then it kind of turned into an argument within the team. And then at the end we just lost the team member, he left the company. And after that they had this higher level meeting, from what we know. And they decided that they now want to be more welcoming to new ideas. And they should be more open to this. I think that's why they were more inclined to accept my improvement. So I think this also shows that the team is learning from mistakes.

Researcher 30:33
So what happens that there is there is your case, I will come back to your case, which was not accepted. Your colleague has left because his idea was not welcomed. And what happened? The architectural problem? Were Did it cause any problems to the software?

Participant 13 30:55
Yes, it caused problems, actually. And after he left, we had to implement the recommendation that he made. And therefore, it was necessary, actually, that we found out later. But again, we didn't really talk about like, we should have done this earlier, or this should have happened before. We just implemented it. And then we moved on.

Researcher 31:27
Yeah. So there was no blame, no regret, right?

Participant 13 31:32
No, no, no one ever blame about colleague leaving when we implemented it.

Researcher 31:36
So tell me about your suggestion, which was rejected based on being over engineered? Can you describe to me a little bit what happened? And what was the idea?

Participant 13 31:50
We, we had this storage, in our system in memory storage. And this is a bit technical, but I'm sure you're also familiar with these kind of things. And then using this could actually, I thought, save us a lot of time storing information on the in memory database instead of the actual database. But then we had to change the structure a bit too much for this and the time that we would save, they thought would not worth it. And I think I also didn't have enough information to support my idea further. Therefore, it just went down. The idea just went out.

Researcher 32:54
Okay, but that was a legitimate decision. I mean, you're all together discuss the decision. And you made a decision because you waited, you waited the cons and the pros. And as a team, that decision was not to go ahead, because the advantages did not warrant to go ahead with this proposition.

Participant 13 33:21
Right. Right. We had a lot of people in that meeting, not just one person. And then they all provided their information, their ideas, and we decided maybe later, but definitely for the MVP.

Researcher 33:40
So how did you feel about that rejection afterward?

Participant 13 33:45
Because I was explained why I was rejected. It was fine. For me, actually. And I also learned by rejection, too, maybe? Because I, like you said, I weighed the pros and cons on my own first, and then when I went there, maybe it's not the best solution. It's always better to hear more than one idea.

Researcher 34:14
Of course, yeah, yeah. And you need to to make your ideas stronger and more sellable some to the team before you put it forward. Right. So but it didn't discourage you from taking initiative in the future. They did.

Participant 13 34:33
Right. It didn't the idea that accepted came after this one, actually. Therefore, I was still feeling I still felt like I should contribute to the team more and then the second time they accepted it, for example.

Researcher 34:50
Okay. Nice to hear. So thanks for sharing those example and move to the next item on On the list, in your email answer you said, so the question says, people on your team sometimes reject other for being different. You said no, my manager pays a lot of attention, hiring people with decent attitude to get the team spirit up. Therefore, there is no discrimination inside the team. So there is a great level of support from management to keep this culture on the team. Right?

Participant 13 35:29
Right. Right, my manager told me himself that she added himself the interview item to the company, the [deleted to preserve the researcher/participant anonymity] Germany and Turkey that they would usually look with for the candidates, they would look if they're technically good. And if they have enough English skills, yet, those were the two items. And my manager said, he also brought this new topic, that's called attitude. Now, one day, great candidates, they also consider how their attitude is. So that this is also an important criteria now, because the reason behind is that they want to the team to communicate nicely, and would be more open to discussions. And you know, like how some people can react aggressively, and stuff, I think during the interview, they also test those features for the candidates. And if they're patient, or, or, or if they are nice, then it's a good indication, because they want the team. To a workplace that's more fun to work with, instead of going to work every day and then not liking it.

Researcher 37:04
Judging from your English, I think they're doing a good job in the interviews.

Participant 13 37:14
You also speak very good English, by the way.

Researcher 37:17
Okay, thank you. You mentioned a very good word, which is often why it is important to your team that they check this criteria of openness in the interview.

Participant 13 37:35
I think before my manager came to the company, they didn't check this. And then because of this, they had a lot of people that are highly skilled, but they had poor communication skills. And because of that, they had problems like the software was high quality, but they want to leave the company also. Therefore, they want to change things around and then make it more people that are open to communication. So that the lifecycle of the software is longer.

Researcher 38:16
Do you think that openness is also important in this environment? Because you need to accept initiative, you need to accept feedback, you need to accept rejection, do you think it is that's why it means a lot to your manager?

Participant 13 38:35
I think so. Because, like I said, we all have problems with development from time to time. And then I think we need to be open also, to help others. Also, to understand if we're doing if we're, what we're doing is wrong, then we should understand it and then change be able to change the way we approach things.

Researcher 39:09
Okay, fantastic. We will move to the next one, which is about taking risk and initiative. We already talked about taking initiatives. I'm just gonna read your answer and see if I can follow up in any of it. So only if the team's knows and approve, you take the risks. So a team member once took a risk and change part of the software for good, but without getting any approval. His work turned out good, but he was held responsible for changing things without notice. That's a very good example. So as a team you have to discuss and make the decision together before somebody can go ahead and implement his initiative or risk right. That's what you say.

Participant 13 39:58
Right? Right. Exactly. So if you want to change something, you have to explain why and, and you have to have an approval to go ahead. It's also important that if they value your idea, they can provide you the time to do it.

Researcher 40:21
So this colleague of yours in this example, you provided the near answer. His initiative or his risk taking, Did it improve the any aspect of the software? Or it didn't?

Participant 13 40:40
I think he improved things, because maybe I can explain it to you this way. Yeah, the software itself is written in Java ecosystem. Just there was one part that was written with JavaScript. And the thing that this colleague did is make that code also written in Java. So every everything is written in Java. And then also, if you asked me if it improved the performance, maybe not necessarily, but what how it helped, is not everyone is comfortable with JavaScript in a JAVA written environment. So it's easier to maintain doing it this way, actually. But like I said, he he first suggested this, it's not that he never mentioned about this, he suggested that but but never had approval. And then he just went ahead and then implemented it in on a weekend, I think. And then. And then he said, here's the code, whether you use it or not, it's your decision. But if you want to it's out there. So it was not the best approach to things. But that's how it happened. And right now, we actually use his version. And but he no longer also work with us.

Researcher 42:17
Okay, so what happened to why he left in the first place?

Participant 13 42:23
It's similar to this other colleague that that he was also there from longer time. He's also very highly skilled. And I think he also had a difficulty communicating his ideas? Because I think, first of all, when he approached things, with good ideas, but with a mother good attitude, I mean, you have a good idea, but the way you say it is not nice, sometimes you know, that I think that's one of the main reasons that these ideas are so rejected. And then I think he also didn't felt welcome. welcomed in the team. And I think he just found a better opportunity. And then,

Researcher 43:22
Yes, I agree with you. In an environment like this, you need to know how to deliver your ideas, you need to know how to communicate your ideas. Because if you fail to do so you may get rejected for the wrong reasons. So highly likely, this is what happened. He had a good idea, but he didn't know how to sell it to the team. Right? Yeah.

Participant 13 43:47
And because I know that this, these are the things that happened in the team, I always pay extra attention to suggest my idea, the way I suggest my ideas to the team. And then when I'm nice, they're also nice, approaching me back whether they accept or they reject.

Researcher 44:07
Overall, it looks very constructive and healthy way of approaching ideas and initiative. And it is normal that some people fail in this type of environments. For many reason, it doesn't have to do with the environment, it's could be because of the individual. You remember how your manager filter in the interview and look for openness? It is important to make sure that the person when he or she walks in, can function can operate in this type of environment.

Participant 13 44:44
Right, exactly.

Researcher 44:46
So, the next one, it's about helping each other. And your answer was, yes, my team is perfect in the sense we always work in pair programming. Everybody asked for how, and we have session to discuss ideas. So do you work in pair programming when you need to help each other or pair programming is is a practice in your team.

Participant 13 45:13
It's common practice in our team, actually. So the way it happens is that during our sprint planning, we decide who wants to take a particular story. And you know, sometimes two people want to work on the same story. And then we just put their names on it together. And then they work pair programming. And in other cases that late, rather junior colleague needs help working on the on the story. And that's also another example where we work pair programming.

Researcher 45:54
Okay, do you think it helps people to improve their way of coding and the quality of their code?

Participant 13 46:01
I think that's, that's one of the best things where software engineers can improve themselves. By just by pair programming, you see a lot of new approaches where you don't see on your own, because you have your own set of ideas in mind how to develop a code. And even if you make research on the internet, you don't see them done in front of your eyes. And then when one pair programming, you just see a lot of things written different than you thought, but or shorter. But you see them in practice, and understand, I think it contributes a lot, I definitely learned a lot from very different approaches of my colleagues.

Researcher 46:45
So do you think it's improved the quality of the code you write?

Participant 13 46:51
Yes, definitely. Because like, I also see the code myself, when I open the ID, EA, and then look at it, I also see that the code is written that way, but I don't necessarily understand or start writing it that way, right away, but when I see someone actually writing it, I'm like, Ah, then that's why they do it this way. And then I think when you see it in practice, it's more understandable.

Researcher 47:23
Okay, fantastic. So do you have an example where somebody helped you, or you help somebody in the team, and it did improve the way they write code or the any aspect of quality.

Participant 13 47:44
I think, me that someone was helping me. And then there was a particular problem that I didn't know how to track, necessarily. And then this is not necessarily like the quality of the software, but they helped me how to use the IDE to better make use of the IDE to track problems. And then I didn't know

about a lot of the things that they provided. And then it definitely helped me with it. And then from then on, I used more efficiently, and then it helped me find and detect problems easier.

Researcher 48:28
Sorry, what's the IDE?

Participant 13 48:31
It's the I don't know, the abbreviation, but it's the development environment. Okay.

Researcher 48:41
Okay, so Well, definitely, it helps you to, to improve your efficiency, but also the way you write code because you use the environment in a better and efficient way, don't you think?

Participant 13 48:55
Yeah, definitely.

Researcher 48:59
So let's move to the next one, which is no one in my team would deliberately act in a way that undermine my effort and say, No, honestly, my team is very supportive instead of undermining, they bring more people in to help in case needed. So how is the team supportive? Can you explain to me how do they support each other?

Participant 13 49:24
First of all, this is also like these are all connected to each other.

Researcher 49:28
Yes, yes. They're very linked. Yeah.

Participant 13 49:31
And let's say when we face a problem, we instead of undermining the person, they just provide more people to help them and then and then we just focus on the problem altogether, and then try and solve it.

Researcher 49:57
So yeah, do you have any example? I think we already discussed that, but do you have any example where you got together as a team? And you solved a quality issue, for example, design issue or any issue that relate to quality? And how did you go about it?

Participant 13 50:17
Right, right. Yes, we definitely got together. I think we were four people. I mean, I have a lot of examples, actually, where we do things this way. Because we work very closely as a team. And when there is a problem, I think everyone is aware of it. And then we say that during the dailies, we say that, so this colleague, if you have any problem, feel free to let me know, I have some free time. And we can work together on your issue. And sometimes those are individual issues. But sometimes, let's say

product owners come with that problem, or testers come with a problem and say that, guys, there's the senior, this serious problem where market is also affected, and we should fix it. And then, and then we all just stop whatever we're doing. Like, and then focus on the problem altogether.

Researcher 51:26
So the problem becomes the focus of the team. So yeah, so you jump in, and you help each other. Right.

Participant 13 51:35
I, sometimes I also ask, like, guys, because of short of time, I have this problem I know you want me to solve, but if you can, let's say if one more person can help me, then we can fix it faster, or when it's necessary. And then, and then they also say that, okay, maybe this person can help you. But he's busy with other things. So we should take off his workload, so that he can help you.

Researcher 52:11
So it's very flexible, people can put their work aside and come and help us to, to push you forward. And to help you with your problem.

Participant 13 52:21
Right. And we do it officially like we take off their stories out of the sprint, and then put it in the backlog so that we can take it later. So it's not like extra work is put on their shoulder. But it's also officially allocated time to help others.

Researcher 52:40
So formally, you're free the person to help, right? Yes, so you don't create an extra burden on him or her?

Participant 13 52:49
Exactly. Because this, this can also happen. And people may ask you to do it out of the route, they may want you to do it without officially providing you more time. pressure

Researcher 53:02
on you. Yeah, that's a very healthy way to go about it. Because the person might be helping and may be feeling that there is pressure because his story is not moving on the board. Right. Right. You talked about this work environment, helping each other feeling safe, it helps efficiency. But do you think that the quality of your work and the quality of the software you're producing it's also improving because of this?

Participant 13 53:36
I definitely think so. Because the colour rate is also about it's on one side, it's about how performant your software is. And on the other side, it's also how error prone your software is, let's say its performance, but also it there are a lot of errors during the lifecycle of your software. So when the team gets together, I think the errors are sold better. Therefore, I think it definitely helps.

Researcher 54:18
So the errors are solved, but are they reduced? Do you think in this work environment you reduce errors as well?

Participant 13 54:27
I definitely think so. We definitely allocate some time during our Sprint's to fix a bug fixing, for example. And then for those sometimes team has to work together or work closely. And therefore it definitely helps.

Researcher 54:48
So you mentioned that you dedicate time on the spring to fix bugs. Do you do that because you feel it's it's Something that you need to do you, you feel it's an obligation or because you have a lot of freedom to make decisions or

Participant 13 55:12
During our sprint planning if we see that there are a lot of bugs in our backlog written by our testers or QA engineers, we just think that it's time that we should allocate more time on bug fixing. I think it happens in once in every two sprints that we allocate time more or more time for bug fixing, maybe 30% of our capacity we allocate for bug fixing or existing in our backlog. And then I think it's a necessity for quality of the software.

Researcher 56:02
So you feel as a team, it's your obligation to clean up those bugs, and to assure a better quality for the software. Right?

Participant 13 56:10
Right. Right, right.

Researcher 56:12
So do you get clashes with the product owner or the management when you make those decision to focus one sprint, or 30% of the sprint on the quality of the software?

Participant 13 56:31
At one point, they wanted me to work almost all my capacity on bug fixing, because the bugs that were collected on the side of the software that were I work on. And they wanted me to work almost only on bugs for one sprint. And then I was not necessarily happy with this decision. And then then and then I said this out loud in our not sprint planning. But in one of our PII readings. I say these words, because I think you're familiar with all scrum methodologies.

Researcher 57:17
So you feel it's safe to say no, right?

Participant 13 57:20

Right. And they said that sorry, Participant 13, we have to have you work on these bugs for discipline. But later on, we will give you the space to work on a part of the software that you want to work on.

Researcher 57:39
Okay. Fantastic. So last question, which is working with member of my team, my unique skills and talents are valued and utilise. You said, Yes, you got hired as a full stack engineer. And you would like to work on the backend. They respect and try not assign in me to any front end related or if you don't like front end?

Participant 13 58:06
I don't really like working with credit. Also, I think I'm not talented. Working with front end or UI related.

Researcher 58:14
Yeah, some people find it boring to work in the front. And yeah, I understand. So they respect your, your, your wishes, and they value your talent, right? They don't push you to do something you don't like, exactly how does it make you feel? Does it make you feel? How does it make you feel?

Participant 13 58:37
I want to go deeper in, in my knowledge on backhand side. And if I was distracted by learning something, I don't want to focus on like front end related stories, that I think I wouldn't focus on my job as good as now. Because I think people are more efficient, when they love what they do. And if you're given the space, or if you're given the direction you want to work on, then I think it definitely helps you put more effort. Because you can also think more creatively if if you're working on what you like. And in the end, it also affects the software code.

Researcher 59:32
That's very true. That's very true and very wise of you to say so. Thanks, Bear. Kai, thanks a lot. And last question. Any things you'd like to add in this conversation about the work environment and how it does it help you and how does it help quality? Something we didn't talk about before we conclude?

Participant 13 59:56
We may maybe I forgot to mention that we Follow I think it's called Safe framework.

Researcher 1:00:03
Yes. It's, it's a way of scaling Scrum or agile. Right?

Participant 13 1:00:08
Right. Right. And we also had, I think, one whole week, like one full week training classes. When we got in the company, we were also encouraged to apply for certificates. Because I think there are like certified safe practitioners. And I personally didn't have the time to prepare for certificate exams. But we were also encouraged to do this. I think, if, before I worked here, maybe I can briefly mentioned it to before I worked here, we didn't, I didn't have like a good example of Scrum practice. So like whenever

scrum was tried to practice in in our environments before, it always looked like waterfall to me. And honestly, I can definitely see the difference between waterfall and like, Brown.

Researcher 1:01:29
I agree at the beginning, it looks like another version of waterfall. So what's the reason? I mean, yeah, it does look but once you get into the dynamic and you practices, you see the difference here, I agree with you. The first time when the idea of us and Scrum was brought. I was skeptical. I was very skeptical. I thought it just another way of dressing up waterfall, but once you work on it, you get the rhythm and you see the difference? Yes, I agree.

Participant 13 1:02:00
Exactly. Exactly. I think when every everything is defined, and what you need to do when you need to do are defined. It's easier for you to focus on work. And I think Agile methodology do this in a very good way.

Researcher 1:02:20
Yeah. And they help you to deliver faster, right. Right. Okay, thanks, Participant 13. It's a Sunday and I let you go. Thank you very much. I appreciate the discussion and the example. Thank you.

Participant 13 1:02:34
Thank you. Have a good day. enjoys my conversation. I hope it helps you with your research.

Researcher 1:02:42
It does. It does many things. Have a good afternoon. Bye bye