Participant 1

SUMMARY KEYWORDS

people, quality, software, performance, question, client, team, safe work environment, blind spot, academic, safety, fantastic, code, practice, work, project, robert, achieve, test, read

Researcher: 00:19
Good morning, Participant 1. How are you?

Participant 1: 00:22
It works. Good morning.

Researcher: 00:24
Well, how are you doing? I'm good.

Participant 1: 00:29
Thank you. How are you doing?

Researcher: 00:31
I'm doing better. Well, thank you. I'd like to thank you for accepting doing the interview. I really appreciate. Thanks for your time.

Participant 1: 00:38
Yeah, thank you for I think you invited me maybe? Yeah, you invite me. Okay. Yeah. Thank you for the invite.

Researcher: 00:48
Yeah, I invited you. Yeah, that's correct. I think we chatted before and I invited you based on the last time we talked. All right. I do have a lot of questions. So I'm hoping to. Yep. So I start by introducing myself, my name is Researcher. I'm an [deleted to preserve the researcher/participant anonymity] University. I do research in the topic of agile and how agile team achieved quality, but in particular, I'm more interested, how the social environment helps them to advance the pursuit of achieving quality. So I have less interest on the technical aspects. But I'm more interested in how people work together and socially how they manage achieve quality. So do you have any questions for me? Before I start questioning?

Participant 1: 01:47
No, I'm fine. We can proceed.

Researcher: 01:52
Okay. Before we do that, can we start with an introduction? Can you introduce yourself and tell me what you do?

Participant 1: 02:03

Participant 1, 33 years old, Dutch I am a software developer lead. I manage a team with five, and four other people. We're all developers. Or more specifically, everybody has their own background. I am usually responsible for maintenance. And I do a bit of guidance, client guidance, so to say. And all that is needed to run a company. Okay, fantastic. Yeah. So like about your background, like, like, I assume this is your PhD, or is this true? No,

Researcher: 03:04
I finished my PhD long time ago. So I've been in the industry for almost 20 years. Yeah, I worked as a software engineer. And I came [deleted to preserve the researcher/participant anonymity] But I lived outside [deleted to preserve the researcher/participant anonymity] last 20 years. I did my bachelor degree in [deleted to preserve the researcher/participant anonymity], and master's degree, and I worked as a software engineer. And after that, I worked as a business analyst. And after that, I worked as a project manager, I delivered few projects. And I decided to do PhD and work in academia [deleted to preserve the researcher/participant anonymity]. Any questions for me before I continue with the questions? So in your team, what methods do you do? Do you follow a specific agile method?

Participant 1: 04:45
I'm not as experienced as you are. I've been in the industry now for Eight years, I think. And you start off as absolutely naive, I can program. And now I've come to the conclusion that it's all an illusion, you have people learning to stay relevant. We use and I have to explain this XP programming. But as it is described on is basically the right way. Because it has no judgement in it. The current Yeah, the judgement that is normal in the in the in the world is like, our programming is just, they're just programming and like, whatever. They're just doing whatever. But actually, if you have a fully autonomous development team that decides everything. The good part is it actually runs way better than something that has any any form applied with any method applied to it. So that's, that's what we try to do. We are developers, and we plan everything together. So so yeah. But as we are also interconnected with the rest of the worlds, whatever the the clients, or the other party require. So that's either daily stand ups. But we tried to steer at the minimal amount of form. Let's call it like that. Yeah.

Researcher: 06:56
Okay, fantastic. I understand.

Participant 1: 06:59
And then also the waterfall method, like you give us like a fixed fee. We take it and we go down under with the submarine. And six weeks later, we present you the project, if the client agrees to that, we also do that.

Researcher: 07:12
Okay. Yeah. So. So you already talked about the team, the size of the team, and you do a mixed of what did you call it that XP and Scrum, right?

Participant 1: 07:32
Yeah. I'm not really like into the depth, probably as you are with terms and constraints and what exactly is. So it could be anything from daily Scrum? Yeah, we do daily commits, basically on all the projects. And if the client wants it, we do test driven development, or we want it but the client usually doesn't want to pay for it. So depending on like, is it?

2

Researcher: 08:00
Okay. Let's talk about the type of project you get from your client. What type of project do you get? What type of software? Do you develop? In general?

Participant 1: 08:12
Custom software. We do it in Ruby, or node, or elixir, or Python. Usually, it's a CRUD app. Usually, it's an API.

Researcher: 08:27
Okay. So you specialize in this type of software? Right?

Participant 1: 08:34
Correct. Yeah. And usually like the, how we get our clients, this client has a problem with an agency. They are looking for second supplier. They get a poor quality from agencies and then they stick with us.

Researcher: 08:53
Fantastic. The questions, I've sent you in the email. The preliminary questions, but we will align them to software quality. But software quality can mean a lot of things to a lot of people. So I'm gonna use a definition and we will discuss it briefly before we go to the rest of the questions.

Participant 1: 09:14
This also educating for me. Yeah, that's

Researcher: 09:18
Fantastic. I'm glad you find it this way. That's that's very, that's very encouraging. So we use we decided to use ISO standard definition. Yeah, so their definition states is sort of ISO, which one, ISO 225 Zero 10. I can send you the link later on if you'd like. They defined quality as the degree to which the system satisfy the stated and implied needs of its various stakeholder and does provide value. It's also state and cover a set of non functional characteristic, mainly performance, compatibility, usability, reliability, security. maintainability portability. So, first, do you agree or disagree with the definition? Or would you like to comment on it? Would you like to add to it?

Participant 1: 10:30
My comment would be, which I see. Currently, I've been like reading in the last two months about incident commanders. Because this was Yeah. Previously, like, topic I was, I AM, basically, but I didn't know that there was an actual role for it. So anything that they have as guidelines? Or, or or? Yeah, so short, I think it's pretty complete. Nothing to add, am unhappy is there because I actually read the 9000 1000 27,000. Yeah, on this level that I am, I just need to know what other people?

Researcher: 11:31
Yeah, the 9000 is the old version. Yeah, this one is the latest version. I'll send you the link later on. Okay. So this is quality. I'm glad we agree to a definition because it can get philosophical. So So what do you do in your team, what type of practices and processes you have to assure quality?

3

Participant 1: 11:59
Okay, I'm just going to be very practical encryption. is the first one. So distinct encryption, I would say, two factor authentication everywhere. I would say YubiKey is the latest thing that I've discovered. That could be something that is like, extra relevance or extra secure, my personal and one older partner are using for their password management, an open source software called key POS. So then you have your personal, let's say, sets setup, then let's say it's over development. Wise. I would say you start with Yeah, let's set let's say, we have a default project. We need Burke monitoring. We need Yep, monitoring error monitoring. So that's the first thing we install or I installed. Documentation, I think is also definitely something that decreases debugging time if you're not there. And then what else? Yeah, tests driven development. Pull requests, coding standards and continuous software development. Yeah.

Researcher: 13:32
So do you follow any software engineering best practices? You talked about test driven development? For example, do you have a CI service etc? Yeah. Can you tell me exactly what do you have? What do you use?

Participant 1: 13:52
Yeah, so that's what we use. We have a CI service.

Researcher: 14:11
Yeah, fantastic. So, by the way we use, we have used

Participant 1: 14:17
Git lab CI CD,

Researcher: 14:18
Okay. I've read your, your preliminary answers, and they are rather very interesting. From the answers to the questions, the feeling I had, that the environment you're working in, is relatively highly safe work environment. I can explain what do we mean by safe? A safe work environment is a work environment that provide a sense of security. What does it mean is a sense that the person won't face any risk or repercussions and admitting mistakes is okay, you feel that it's okay to take initiatives, you feel it's okay to discuss problems, there is a sense of confidence that the team will not embarrass or reject you based on your opinion or speaking up. This confidence stems from the mutual respect that exists in the team. So when I look at your answer, it looks like it's highly safe work environment. Do you agree with me or not?

Participant 1: 15:39
It has both both sides.

Researcher: 15:41
Okay. Tell me about the both sides

Participant 1: 15:48
Is it a safe work environment? I would say it's a safe work environment. So you have to be alert. We are very transparent. You're paid on results. You get compliments if you do something like exceptional. And people are at large are very direct if you make a necessary mistake. But obviously, like, yeah. So so you get used to just working within your circle of competence and not going outside of it? Yeah. So So

4

like, for me, I would say like, it's very performance oriented organisation. And that's Yeah, I think safe, because if you're not performance oriented, it's unsafe, because people are busy paying attention to non relevant things.

Researcher: 17:02
So in a scale of five, strongly disagree, disagree, neutral agree and strongly agree. How How much do you agree that it is a safe work environment? Like you said, based on performance?

Participant 1: 17:17
So from the scale is from minus five to plus five?

Researcher: 17:22
Strongly disagree, disagree? Neutral? agree and strongly agree?

Participant 1: 17:28
I would agree.

Researcher: 17:29
Okay, fantastic. In your opinion, what's what's has promoted this culture of safety in your work environment, which is relatively safe?

Participant 1: 17:46
Safety? This is socially like sake, what what is yes,

Researcher: 18:03
Yes. It's admitting mistakes is pointing out mistakes to each other, you know.

Participant 1: 18:11
From leadership. Yeah.

Researcher: 18:13
So the leadership promotes this quality, right?

Participant 1: 18:17
Yeah. So we have like, I would say, two exceptional humans within our team. One is a more academic person who is like, yeah, he promotes the right values. And the other one is a very pragmatic, like, yeah, he basically you give him specifications, and he's already delivering. So this is the combination of the values that gets promoted. And we always argue like, Okay, should we actually like to be striving for the values? Or should be striving for that delivery? Like the industry, the commercial side? And this is always the most dense? Yeah, describe conflict we have. Yeah. So So what are the best practices? I would say? Yeah, almost open conversation. Extremely direct. Yep. To the point. Yeah, clarity. Well, document well documented so that everybody knows like, Okay, what's the decision that has been made here? So transparency? Yeah. Okay. Like we are like we are basically yeah, we can say like, like, I like reading about how good luck does it.

Researcher: 19:44

Okay. Let's move to the details. Open. So and the first question I asked in the email, if you make mistakes in your team, it is often held against you. So your answer was yes and no. If you make mistakes, that's because of a bad practice than people will call it out. And if it is a best practice than people won't call it out. So can you elaborate a little bit on this answer?

Participant 1: 20:24
Okay, so where a team member can follow a good practice. For example, somebody pulls the database down, he used the wrong query. He reported it to me like, yeah, Participant 1, you need to restart or look at the server. Sorry. Yeah. And then yeah, basically, I have to say, like, like, that's just not what we do. Right. So so this was like, basically him, I think, without a coffee. just assuming that the query was right. It wasn't an what we? Yeah, I would say like, we pull the production database to local, we run the query there. If it works, great. If it doesn't, you crush your own machine, whatever. So yeah, and he instantly like, either he apologised, or he said, like, sorry, sorry, like and reported to the client and like, yeah. So that's something like, it's almost like you, don't you like? Yeah, you don't want to embarrass yourself, but you just uphold the standard. And he's very proud. So.

Researcher: 21:59
So let's let me follow up. Do you think he felt safe to do so? And to admit the mistake to you? So do you think he did so because the environment allows him to do so like you said, the values of the team is

Participant 1: 22:18
Yeah, absolutely. Yeah. It's a bit of like, his design the environment. So of course, they this is just kind of practice. And it's also like, if so, for example, he sets the standard and then if he doesn't, like conform to is understand that yeah, that's a bit like, a shaman to ashamed of. But that's all good. Because that's all part of the the culture we are in. Yeah.

Researcher: 22:50
So, so you also said no, and the answer, correct.

Researcher: 22:55
Okay. So so when it was not held against him or her,

Participant 1: 23:04
We use century as error monitoring tool. There are two things that it's currently monitoring, I wasn't aware of this, it's errors and transactions. And I allowed apparently 100% of the transactions to be recorded. I didn't turn this off. Century way over budget, like, way over budget. I had to report it. But of course, like Yeah, nobody knew this, this configuration, or nobody knew. Like, they didn't read the sensory setup. So they're like, that's fine. Like so this is like the let's say the if you know, you do bad things like that if you're not an integral person, basically. Yeah, that's something that's not like allowed but if you just do stuff because you're naive and stupid, like like, you're just don't educate yourself beforehand yet.

Researcher: 24:06
So what's happened in this in this instance? You got blammed?

Participant 1: 24:10
I informed the rest okay, what they should do set

Researcher: 24:15

Okay. Okay, fantastic. So this climate of accepting and being transparent about errors, how does it overall helps the quality of your software?

Participant 1: 24:32
Versus also patient?

Researcher: 24:35
Should I repeat the question?

Participant 1: 24:36
No, no, I'm looking if it also was a question because then I can already like, yeah, okay, so So how does it help us? Well, for example, we just besides the software, like software just needs to work. Does it work yes or no, but all the stuff around it impacts either for Velocity of delivery or velocity of repair? Yeah, it's just improving improving your game. Yeah. How does it help? Yeah. How does it help? Yeah. In general, like, like, you can, you could do the same thing a million times. But yeah, that doesn't really help. Yeah, I think that's the like, we are very good. It's roughly driven. I don't know what you want. We're sorry.

Researcher: 25:34
So let me put it in this way. For example, when people brings errors, you help each other to correct and the process gets better. Yes. Is that what you mean? By iterative?

Participant 1: 25:47
Yeah. Okay. So yeah, so our default is to help each other. So let's, I think that's maybe not in the industry, something that is so normal, like maybe somebody are, it's like, if you don't know it, like, you should figure it out yourself. Like, it's not my responsibility or something, we have a shared responsibility. And this is, I think, our strength. So if somebody has like, four, it's always difficult to say, like four hours, eight hours, 16 hours of personal debugging, and it doesn't work. Yeah, you just due to the internal Stack Overflow q&a, you just write down what you did, to be able to reproduce the situation. And then you do pair programming. And it's even, like, I just recently seen this, like, that. Also, juniors can help seniors with with pair programming. And this is amazing, because I'm like a junior mentor compared to the on the older guys. But even like, the academic guy was like, Participant 1, I know, like, your skill level is like, just above like zero, but I just need somebody to like be a rubber duck. So sit here and don't talk. And I was like, yesterday, cannot already see the mistake. And he was like, okay, okay, this is too much for me, you can sit down. And so how does it help? I would say like, yeah, finding blind spots in it. Yeah. It's just like, that's what we do, basically. And how does it help? Yeah. We are very focused on just the result. That's what we care about my personal empire building our stuff.

Researcher: 27:35
So this focus on the result and helping each other. It allows you to improve as a person, and do you think subsequently, the quality of the work gets better?

Participant 1: 27:50
Yeah, yeah, that's an inherent. Okay. Of course, we have our own issues with no, my code is perfect and whatever. But yeah. Like, it's the other side, there are big egos. So sometimes you have to like decide, okay, who's going to take charge here? And that's the only time that there's like conflict, like when there's like a renegotiation of the commitments. People change and the bad behavior is corrected or improved.

Researcher: 28:27
Okay, I'll move to the next one, which is a member of your team can bring up problem and tough issues. To discuss. You said yes. The most reliable way to run software is to have a safe environment, where you are to bring up problems and tough issues. Do you have this is very interesting. That Do you have an example of how bringing up problems helps the quality of your software?

Participant 1: 29:05
Yeah. Okay, so

Participant 1: 29:18
Yes. So So an example where a team member brought a problem related to software quality?

Researcher: 29:25
Yes.

Participant 1: 29:28
Yeah. Default like, like literally the only thing that they bring up is we should be automating testing. Yeah, that's, and we already have that. So we already have the pipeline. We just should be more testing. Right. Let's be the first one.

Researcher: 29:56
So automation has its make the process more efficients I guess

Participant 1: 30:04
More coverage? Yes.

Researcher: 30:05
The coverage? Yeah. So you keep adding to the automation suite. And the more scenarios you add, I guess you find more defects or the the more coverage you have of the functionalities, the less defect or you catch more defects. Yeah. Okay, fantastic. Yeah. Let's move to the next example. Bringing up problems, how does it help problem related to follow in or adopting quality assurance practices or processes?

Participant 1: 30:41
Yeah. So we should really do maintenance of end of life applications. And then more often. We have in this business now, for six years, we see a, let's say, a handful of applications being end of life, either. Yeah, whatever. And it's my opinion, that is also currently being being discussed, that we should be more aware of these end of life scenarios, because it impacts our let's say, our business. And yeah, as I don't have any experience with actually successfully migrating from from these, these, these versions, I just want to know how to work. Like I've been always in been dropping out features, but maintenance is a whole nother ballgame. So currently, we're discussing the topic.

Researcher: 31:55
Okay. Another example is also related to people bringing problems, how does it help people to adopt and follow best practices?

Participant 1: 32:12

Is that the last question? Yes. Okay. Yeah, I have here, we should really do backups. Not sure if it covers the question, to be honest.

Researcher: 32:28
Yeah. What type of backups? I mean? Yeah, can you Yeah. Can you give me an example?

Participant 1: 32:32
We have? Yeah, of course, like, like, and this is my limited awareness of the owner, what a software developer usually does. That's my point of view. Let's call it like that. But what we do is always business critical. So we have infrastructure, like I am on so many service level agreements. It's not even funny. So yeah, backups should be there. So it Yeah, and we should practice rollbacks. I'd never practised rollback and what I mean by backup is not only the files so that's like the contracts the documents, the images, whatever. It should be databases, it should be the whole application it should be the whole disk it should be whatever ecosystem is set up product point.

Participant 1: 33:52
Yeah, and backups is definitely a process I would say is like like Yeah, even the tax authority has like their own setup for this. So we should also be doing like practising this

Researcher: 34:08
So this this practice helps you to recover if the case of a failure right

Participant 1: 34:16
Yeah, and that's basically quality. Yeah.

Researcher: 34:19
Okay, fantastic. I'll go to the next item which is people on your team sometimes reject other for being different and you answer yes. If the other person is clearly not now negotiable now, if the other person is knowledgeable. This also relate to what you mentioned before. This safety in your workplace is also balanced with performance right?

Participant 1: 34:48
So we are Yeah. I don't know if this is like irrelevant, but If you're not knowledgeable, like, go away there are a lot of magnetic knowledgeable people out there in the world that want to be in sofar world and want to add value when want to bring run a business and they can raise money, whatever, how many Euros they think they need. But to be in business also means to be spending time with people that you actually like. Yeah, expose yourself to. So we select both freelancers that we Hooray our end clients. Yeah. Very, very selectively. This is more internally, right. Yeah. Team. Yeah. So team. Yeah. Yeah. Like, for example, we've started to to look abroad for hiring remote work. So that's a global talent marketplace. Yeah, you know, already. And, yeah, we, we see people that say, like, I can speak English that are Russian, and don't speak English. We see people do you have strong Wi Fi connection. And they're a lead for you, like, between their grandparents with the guests? And they're like, yeah, we can do this. And we see. Yeah, I'm, yeah, whatever. We have seen everything, basically. And the only thing we look at is maybe five minutes of code, go run through your architecture, and then it's a higher or no, and when can you start, like that's, that's our hiring process. And afterwards, if we have found the competency, and we found some some really good people, we can laugh, and it's completely safe. Like, people shared a whole life with us in the in the internal slack. And we want that as well, to know them. So but competency first, like results first. And we hire like entry for polyglots. Like like really tough to crack.

9

Researcher: 37:04
So rejection in here your interpretation is performance. If you don't perform well, if you are not knowledgeable, then we will reject you. If you are knowledgeable than you are welcome to our safe work environment. Alright, so do you have an an example? I'm more interested in knowledgeable scenario because it fits within your culture of safety? Which is good. Do you have an example where somebody is knowledgeable and he or she brought something to the team's attention regarding quality or quality assurance or best practices? And how it was received by the team?

Participant 1: 37:54
Yeah, it's just like, we value quality so so I don't even perceive quality as as a separate item. It's just that Yeah, with as an entrepreneur, as a human, you have a limited amount of attention. So you can't cover all your, your your basis

Researcher: 38:22
Yeah, so you cannot cover your basis you need to collaborate you need the help of your colleague to see the the dead spot like you said before, right?

Participant 1: 38:33
Yeah, so for example, practical, more practical. Exactly. So we have a very auto detected care person. Our team never enjoyed any education. He just like, sits at home yanks computer, or seven computers, whatever. He's the DP the Yeah, the arch nemesis of the academic who's like value based principle based very abstract, everything should be doing everything. gun like that. The only battle they have is naming basically, like naming stuff. And that's quality already.

Researcher: 39:13
Okay, great. So, um,

Participant 1: 39:17
Like, the because it starts with naming like, What the fuck, like, like literally like, like, who is like, you don't understand anything of this, like, tomorrow or a week? And then the other person? Yeah, but it works. Okay, fine.

Researcher: 39:32
Yeah. So if it's not readable, the next developer might not understand what he does. Might not be able to maintain it. Right.

Participant 1: 39:40
Exactly. Exactly. Exactly. Yeah.

Researcher: 39:44
I understand. I developed software for years. Back then, maintainability and readability was not a priority for a software

Participant 1: 40:00
What do you it needs to be done yesterday, like, it doesn't matter.

Researcher: 40:05

So the priority was to deliver and to get a software that function that most of the software we built those days back in the 90s. It has to be decommissioned after five or 10 years, because it couldn't survive. The continuous evolution that a software needs. I know what you mean. Yeah. And I'm happy to see people like you with eight years experience already value this, this quality,

Participant 1: 40:36
I value it, but let me let me be. Let me be crystal clear. My my influence isn't yet there. Within the within the team, like it's still like should be done yesterday. So yeah. And also now like like, this may be like one minute sidetrack. We're building like a quality all the things that for

Participant 1: 41:01
Based on the ISO 9000. For the aerospace, world, industry. I making it and it's ugly, the goat, but it works. And this is like as an intrapreneur. You have to decide like, am I going to be like eloquent elegance? Like, of course, I can beat it as well. But when you're in business, you need to make good good corners.

Researcher: 41:35
Yes, you need to bring the money to keep going. Yeah. Understand what you mean. Yeah. So the next one we already touched. But it's interesting, too, because it's a little bit different. So in here, we looking for a risk taking or initiative taken within the team? And you say yes. And you seem to be having a big yes, yes. See problem. And through issues? Do you have an example for me? So can you prepare an example where the team member took an initiative to improve the team quality to achieve high quality?

Participant 1: 42:02
So that's basically the question. And I think this is also where I started with with the open question before, use an open source password manager, I think raising the awareness about passwords, and how you manage these, and what is actually in your password manager. And being aware that these companies are commercial, and not some some foundation that is like there forever, is a good way to because like I'm very young, as well to the doctor and not really. I don't have any formal education in this in this topic. What else? Big, good Berg reporting software, I think like the Yeah, this is just like, you need to be, like, aware of what you're actually selecting to report the books to you. Like, it's insane that some people just like, don't really pay attention to this. But this is like, yeah. Yeah, the effects are everything basically. It isn't maybe more commercial, but be approved for century like just for a year rotation money. Force disk encryption and force tests, and four is hard test coverage. So get doing the basis doing the basics. And, yeah, we we try to educate customers on this. So we only, like our constraints are what the customers budget is. So we try to really advise the customers to test software.

Participant 1: 43:54
So that's a challenge when you don't have the buy in from the customer in this type of a service provided customer. Because the customer can slow you down. Right? If you don't have a buy in from the customer. 100% Yeah. So do you try to get the buy in from the customer to support you?

44:14
Yeah. So this is just a quick brainstorm, like, I would say below 15,000 euros or 20 25,000 euros. T Diddy doesn't have any value is just like just look at the software above 25 It starts to become interesting, I would say till 75,000 Because then the scope of the project is just getting like, just a little bit larger and like, starts to begin to become like a house of cards. And then I would say above 75,000. So that would mean like, like, like three months, four months of work, maybe? Yeah, then you really need to get tested. coverage because those projects usually continue. And when it continues, like like

when there's no cap basically on budget, or there's like a recurring budget, test coverage is basically a necessity, because you're going to move parts of the software that have huge consequences.

Researcher: 45:21
And this balancing between the risk of the software and the need for quality, you need safety and the teams to be able to achieve it. Right.

Participant 1: 45:37
So let's say So so how we judge this is like, is it commercially like recurring that safety for the company, then? Yeah, and if their safety for the company, we can like already implement test driven development, because we know that this pays off? Like, all documentation is like, if it's forever start testing immediately. Okay, and then the safety is Yeah. Basically, if if the budget is infinity, and the safety is infinity. So So yeah, that's the so the our biggest projects, or the best code bases we have?

Participant 1: 46:27
So, yeah,

Participant 1: 46:30
that's a very fair point to say, because we're like, we're literally working with famous entrepreneurs in the Netherlands that look at the same code that we do. And they're like, Okay,

Researcher: 46:41
Fine. So what, what makes you good at achieving the best code?

Participant 1: 46:48
Yeah, the basis again, like honesty, transparency, everybody can read it. So it needs to be good. Like, it needs to be performance, because the impact is huge. Like it national. So there are, yeah, the, I would say the, it's usually about Postgres, so we have Postgres performance. And like, the academic one is the postgres performance guy. And then the API just needs to handle like, insane, insane amount of data. So that's the other detect he does that. And then my thing is just like, Okay, are there any things that we need to worry about? Like, what things? Yeah, and would that be fine for him? And then how this is like the, I would say the safety is just like, if we need more hands. We hire for performance, like, yeah, that's what we do. So we don't know where to put like, there's no like, like, if the budget is infinity, there's no discussion about performance like zero.

Researcher: 47:52
So even though your values promotes honesty, transparency, safety, you don't compromise performance.

Participant 1: 48:01
No, no, no, no, because it's not possible. It's absolutely not like, like, we wait for the right person to be like, so So I've hired basically, now a military guy, or foreign military guy who only does like security and performance, because that's the only bottleneck we I see right now. We do also financial transactions, like for software, and this just needs to be like, yeah. The highest standard possible. And this is the guy that that actually only does one thing. So I'm very confident we have a good team. Yeah. Yeah. So at first I hired for like, Okay, I want quick, like more team members to to be able to handle the load. But then I really saw like, like volume, or quality isn't really a metric, like people felt you need quality because like we are We preferred small team. super high quality, then. Yeah. 50 developers or something? Yeah.

12

Researcher: 49:09
Great. That's brings me to the next question. So you're performing well, and you provide an environment of honesty and transparency, which means to meet equals safety. So people, do they feel comfortable for asking for help? You said yes. And no. If you have not done your homework, which I think is performance.

Participant 1: 49:39
Yeah, it's a it's a it's a Yeah, if so, if you just ask without thinking, like we actually have this in our marketing text, like we have the text, first thing. Then then programme, which means think about the risks as well. Thinking about the upsides of what you're going to do like, like actually make an informed decision. Because tomorrow, other people will be working with you on the code that you wrote today. And the code, currently, I see the code that we're writing for for one of the national. Yeah, one of the clients, that will be definitely there for 10 years. Or more people will be building on top of our infrastructure. So it needs to be like, good. Let me look at the questions. Sorry, I'm really Yeah, so homework done homework not done. Like it's like, if it takes more than five seconds to explain something. You've not done your homework. Like, that's, that's maybe too high of a standard. But this is what we strive for, high standards. To meet these high standards, we have to help each other's.

Researcher: 51:58
So what'd you say in you prefer that the person takes initiative and try to figure it out in their own before they ask for help, right. So let's, let's work with this scenario, I came to your team. I'm a junior developers, and I couldn't understand a piece of code. And I made an effort to understand it. But I felt like after half a day for hours, I felt that I'm not performing. So I came to you, Participant 1, can you help me? So do you think me Adam, a new member, a junior member of your team? Would they feel safe to do so to approach you and ask for help?

Participant 1: 52:39
I think it's quite clear from the example that we set ourselves how the culture is, meaning, if you've actually done your homework, there is no feeling of unsafety because you already know what you can expect. So you post a message with actual situation, observed situation or the actual observed situation, the expected situation steps to reproduce, you posted in GitHub or somewhere else. You schedule, perhaps even a time with me. And we're gonna sit down and pay attention.

Researcher: 52:54
Yeah, yeah. So

Participant 1: 52:57
We laugh about it when we fix it. Yeah, yeah.

Researcher: 52:59
So this problem solving and helping each other how does it help quality for example? So do people bring for example, when I cannot do a good, elegant code? Do I come to you and ask you do you think it helps quality in somehow?

Participant 1: 53:17
If you're if you can do elegant code, then you're not performing right?

Participant 1: 53:24
If you don't programme then there's no hope there's no Yeah, so I think the like programming is not a one man job. Let's let's let's let's there are they are vocal online. Like like the indie developers, sort of the single developers or or the one one company entrepreneur types. They are like they are there. But I think when you're not at a level that you can be fully autonomous also commercially, you have to collaborate. Collaboration is hard. I think this is raw conflict. Like like this is because other people's code as hell like like working with other people like yeah, other people is not not fun. So you have to be clear on boundaries. When you talk to each other, how you talk to each other, what the jokes are, that they're not serious. Which is hard.

Researcher: 54:37
Yeah, it is hard to create an efficient in the same time efficient and the same time safe. collaborative

Participant 1: 54:49
Work. It's like It's like the being in sync. Realistically, it's impossible to achieve Like literally like, like, because there are always like, like, or you will have, like, let's say there's always million projects that are going on, like code bases, context switching, and naming whatever wrenching whatever. So to take the time for somebody else, yet it's hard. Hmm.

Researcher: 55:24
Fantastic. We're getting close to the end. And we do have five minutes. So your teams doesn't undermine your effort, which you answered Yes. And you enjoy even getting paid for the result? How does this make you feel as a member of the team to be rewarded for your efforts?

Participant 1: 55:52
I would say first, like, if you if you help somebody, okay, so So let's say you mentor, a senior or a junior, or, well, first, you can actually improve their, their, their, their, their debugging skills, which I think is like the highest thing you can like, give to somebody. And then you also get paid for it. So it's both the personal report and the commercial report. And perhaps you get a bit of, like, smarter about the domain.

Researcher: 56:23
So does it motivate you to do quality work?

Participant 1: 56:30
Yeah, I think like, I actually started teaching like, or I started, like, sharing what I know about debugging. More recently, also researching academic work. And so yeah, I just enjoy like performing better, I would say like, yeah,

Researcher: 56:54
Okay, fantastic. I won't go to the last question, because it's very similar to the previous one. So I'm happy with that. So before we conclude, there is anything you like to add to this topic? So feeling honest, transparent, safe work environment? And how does it?

Participant 1: 57:17
Yeah, I would definitely know that I would like to know the blind spots, or maybe maybe the so. Or more academic, like, like, like microscopic food, like, like, I just have it like, because it's my environment. So I am like, like, as in with emotions, like I am the emotion I experienced the most No, I am really like, I am angry. So So I am a team lead. That's my that's my worldview right now. Or like, I am trying to be like,

something like that. But you have a more, I would say clinical academic, like. Yeah. Eloquent, I would say because people that are in academics use, they think about before they write something down. So or, at least I still hope that is. So So your point of view, or maybe your your the phrases that you use, or maybe the stuff that you write about? I would like to also like, Nope, no more about that.

Researcher: 58:21
Yeah. So you mentioned few things, which I like you mentioned the blind spot, I'd like to have the blind spot quite often, in this conversation. What is a blind spot?

Participant 1: 58:36
So for example, you have a person, and the person is convinced about that they are performance. But the client, for example, you have the ISO standard, I think that's absolutely. Like, I need to reread it again, like it because it's just so so well written. Implied something, which implies functionalities, or, like, it's usually not like the code, it works, but it's like, what else? What else did the client expected to do? Like the value which you generate? And that's something I would say, like so so increasing their awareness of what the stakeholders want?

Participant 1: 59:25
I would I would say that's Yeah, but that's just limited to and then the blind spot Yeah.

Participant 1: 59:32
Because you are so convinced that you have done a good job because the software works yet is maybe not what the stakeholders to think. Like they are like, we expect this to work. But the blind spot the blind spot, I would say like people have coding styles, like not everything is documented. Yeah. But generally, awareness like Like, like, like the moral wearing a shoe has about like what the stakeholders want and don't want, the better the quality is.

Researcher: 1:00:06
Okay, Participant 1, thanks a lot. We came to we we made it exactly in one hour. So I'm very happy. Yeah. So what we do is we interview a lot of people and at one stage, we will have a result. And we invite some people to participate in a focus group to understand or to discuss further the results. If you happy, I'd like to invite you to participate in a focus group. So you know, our academic work takes a lot of time. So I'm planning we may have something in June if you like to participate. I can invite you around that time.

Participant 1: 1:00:49
Yeah, I don't know what the what the what the expectations are. So so if there's more information on the Yeah. I don't know. You don't know what a focus group is. So

Researcher: 1:01:02
Yeah, you just show up like today in an interview and we will ask a questions and you will contribute. That's it.

Participant 1: 1:01:10
Sounds amazing.

Researcher: 1:01:11
Okay, then I'll get in touch. I'd like to wish you a good weekend. You too. Okay. Bye bye.