Participant 5

SUMMARY KEYWORDS
team, project, application, discuss, quality, implement, management, security, reject, mistakes, stakeholders, performance, team member, work, software, deadline, people, initiative, agile methods, solution

SPEAKERS
Researcher

Participant 5

Researcher 00:18
Good morning Participant 5. Are you I'm doing well. How are you this morning?

Participant 5 00:30
I'm fine. Thank you. How are you?

Researcher 00:32
I'm doing very well. Thank you. I think we should start because I do have a lot of questions and things to get through. That before I do that, I'd like to thank you for the opportunity to do the interview. I appreciate and I'm looking forward to the discussion.

Participant 5 00:53
Sure, please proceed.

Researcher 00:56
Can we start with an introduction? Can you introduce yourself briefly, mainly your education and your experience?

Participant 5 01:05
Well, My education is software engineering. I am a professional software engineer. I graduated from [deleted to preserve the researcher/participant anonymity] University in computer science and engineering. I'm working in a software firm called [deleted to preserve the researcher/participant anonymity] I'm working there for the last five and a half years. I started there as a senior software engineer. And since then, now I am working as a senior software engineer, team lead for my company.

Researcher 01:43
Okay, fantastic. Thanks for that. We will move to the next questions. You using Agile at the moment, right. Sorry, are you using Agile methods? Right?

Participant 5 02:00
Yeah, I'm using Agile method.

Researcher 02:03
What type of methods are you using? Are you using Scrum?

Participant 5 02:08
Yes, we are using Scrum.

Researcher 02:10
Okay, can you take me through the process? How do you use Scrum?

Participant 5 02:17
Yes. Well company. We have a member of 10 to 12 members. So we had a very small team, we didn't have any complications to make in our project. But after that, our company was growing bigger and bigger. Currently, we have 60% more staff in our company. So day by day, it was very difficult to manage the project. So we thought how can we make it easier to maintain our project, then we were very interested in Scrum. So what we do now we have, we try to maintain it strictly. So we have a scrum team. We have a manager, a product owner and three to four people for each scrum team. And every day when we start our work, we do a stand up meeting. Then after scum, we go to our projects. And when we finish our days, we do a small stand up as well to deliver to discussion the deliverables.

Researcher 03:45
So?

Participant 5 03:47
Yeah, one more thing. Beside that we use project management tools like JIRA.

Researcher 03:57
Okay, great. So how long are your Sprint's one week, two weeks?

Participant 5 04:05
We have currently the project I'm working in, we have a sprint for two weeks.

Researcher 04:13
Okay. How big is the team? How many members of the team are there?

Participant 5 04:20
There are four. There are four developers to QA team, to key remember a project manager and the product owner.

Researcher 04:32
Okay, great. How long have you been working together as a team?

Participant 5 04:42
I'm working with this team for last one and a half years.

Researcher 04:48
Okay, so you've been working together all together for one year and a half right?

Participant 5 04:53
Yes.

Researcher 04:54
Okay, fantastic. So what type of product or software do you develop?

Participant 5 05:00
Well, we developed software for the government filing system called Ino to where we try to digitalize the government procedures from analogue to digital out so that nobody have to use papers. So that that that process could be made much more transparent and faster.

Researcher 05:28
Okay, that's an interesting idea of developments. Thanks for that. Before we go through the question I sent by emails, I'd like to clarify something. Because as I told you, in our email, we will be talking about software quality, and the rest of the interview, and I'd like to agree on the definition and get your opinion on the definition we use in so we use ISO standards definition. I'll read it to you and we can discuss it briefly before we go into the questions you answering the email, okay. Okay. So, the ISO definition says software quality is the degree to which the system satisfied the stated and implied needs of its various stakeholders, and thus provides value. The ISO model also covers some non functional characteristics of the software that the software should address like performance, compatibility, usability, reliability, security, maintainability, and portability. Thirst Do you agree or disagree with this definition? Would you like to add something on it or comment on it?

Participant 5 06:56
Oh, no. Yeah, it's pretty much it. First of all, we mostly work to satisfy our stakeholders needs. Yeah, I agree on that. Yes, the more project mostly depends on those values you mentioned. Yeah.

Researcher 07:14
Okay, great. That was good and quick. So, so, what quality assurance practices and processes you have in place to assure quality?

Participant 5 07:28
Well, first of all, we made a product backlog and we try to make it clear as much as clear as possible, because if the requirement is not very clear, then there might be damage it was it is very difficult to implement. So, after we clear plan, after clarifying the requirements, we usually make some test cases to confirm it, and we discuss it with the product owner with the product owners confirm that yes, if this scenario or this picture, pass this test cases then we will call it down. After development, we pass it to the QA team for more tests and security tests. They write those test cases. Then they checked it, they check that and if that feature passes the test case will hold it down.

Researcher 08:32
So do you have

Participant 5 08:33
So our developers write unit tests.

Researcher 08:41
So do you have other practices like coding standard code review?

Participant 5 08:49
Yes, we tried to follow PSL standards for PHP, okay.

Researcher 08:57
That's great, what those are the processes to assure quality. How about software engineering practices, best practices, do you follow any best practices like continuous integration, static analyzer etc.

Participant 5 09:15
Yeah, we try to maintain some software principles like solid principles, and we try to integrate and precise object oriented programming concept to. So, our company sometimes and this training sessions to maintain code quality. And like, like when we start a project, we, before starting that we set some rules for this, and what can we do what can what we want to do? The setup of quality criteria. Then we start a project and it works as a guideline for us.

Researcher 10:00
Okay, great, thank you. Thanks for sending some answers in the email. I looked at the answers. And it appears that the environment where you work or the software development environment is relatively safe. I will, I will, I will explain what do we mean by safe? And I'll ask you to comment on my assessment. So safe means that the work environment provide a sense of security from repercussions. So people working in your team feel that it's okay to admit mistakes, you feel that it's okay to propose initiative and discuss problem. This sense of confidence that the team will not unwrap, embarrass, reject or punish someone from speaking. There is a mutual respect and trust amongst the team. So when I look at your answer, I think is relatively safe. Do you agree with me or not?

Participant 5 11:10
Yes, I agree. But I think it takes a long time to adjust with a team. Everyone thinks their different ways. Everyone has their own opinion. So it takes sometimes to adjust with each other. But after having a good understanding among each other, I think I think this is a very, very safe, very safe environment. And I feel confident to work in it.

Researcher 11:35
Okay, great. So in a scale of five from strongly disagree, disagree, neutral, Agree or strongly agree? What do you think so what is the level? how safe it is? Do you strongly agree it is safe? Do you disagree?

Participant 5 11:57
I think I will make it four out of five. So yes. So strongly,

Researcher 12:03
Strongly safe work environment. Okay, great. So what made this work environment safe? For example? Is it because of the leadership? Is it because of the way the team work together? So what's values? Or what are the things that happened that makes this quality of safety in this work environment?

Participant 5 12:26
Well, firstly, I think very important is communication. We have a very, very good communication with the stakeholders. The product manager who works with us, we work with stakeholders, we have a very clear and open communication. And if we face some difficulties, and we if we explain properly to the stakeholders, I mean, product owners, they listen to us. So I think this is a very, this is a very good attitude, because in our world, sometimes we face many difficulties, we also have our limitations. So it is very important to understand each other and be open about everything otherwise, there could be some tensions. Also, supporting us which is very good. And the team is very responsible and we help each other.

Researcher 13:26
Okay, great to hear. Let's go through the questions I've sent by email one by one and discuss some of them. And hopefully, we can discuss some of examples.

Researcher 13:47
Yeah, I'd like to hear some examples from you and discuss them with you. Is that okay? Yes. Okay, thank you. I have some of them in front of me and I will read your original answers and you can comment with examples. So the first statement says if you make mistakes on your team, it is often held against you. You said in your answer no. But sometimes it does management has tendency to avoid agile methods, they pressurize us to perform and schedule tasks within a shorter period of time as a result mistakes occur. And they try to blame the teams and the blame for that or the individuals for that. Can you comment on this answer before we work with an example?

Participant 5 14:43
Yes, when I join the team, the management had great control and influence over what we do, you know. They always try to get us to perform on a tight schedule. They were always looking over your screens. So I found it very disturbing, but the good news is that my management is now feeling very much comfortable working with us and more understandable. They try to listen to us so, and now we have an open and transparent communication. When I joined the company, I faced a lot, a lot of struggles.

Researcher 15:23
So this interference from the management has changed what's made it change because you start talking to each other?

Participant 5 15:32
We start talking to each other openly, we also maintaining transparent communication and we stick to scrum and agile frameworks. When they decided to adopt scrum, they also realized that it is good for them not just us the development team. If we work in a structured manner, the output gets better and the quality increased.

Researcher 15:56
Okay, great. Let's work with the example can you share with me an example with somebody made a mistake in this environment and this team of yours and what happens? Especially if you have an example related to quality, how he or it could be yourself came about it, how he talked about it, and how did the team dealt with it?

Participant 5 16:24
Okay, yeah. I have an example explaining this, when I first joined the company, I was assigned on a project, yeah, which has a tight deadline ahead. But I felt that the deadline was very short, since I knew and there is a lot of dust over the requirements. So, what I did was I rushed developing the features, but as a result, when you don't do the analysis of the feature correctly, sometimes you miss some worse case scenarios. Since I was new, I did that mistake. So when the project was deployed in production, we found a major bug, though. So the management blamed me it was a very bad experience for me, but I have too many challenges. My project manager then he stood up for me, he went ahead and defended me and he explained that our deadline was very, very tight. He will not identify those bugs as my fault.

Researcher 17:37
So how could you

Participant 5 17:39
Now things have changed. We try to estimate ourselves the tasks so that we don't have to face the stress and compromise quality.

Researcher 17:47
So how did you feel to when you were blamed?

Participant 5 17:54
Well, it felt pretty bad actually. Because I thought that I did my best. Still, I got blamed so it feels worse. It's pretty bad.

Researcher 18:11
So did it demotivate you or what was your reaction afterward? Or?

Participant 5 18:20
Yeah, it was it was very demotivating. It was very demotivating, but my project manager consulted me, explained to me. He also talked with the manager, then it I felt, okay. I hope it will never happen again. Because I kind felt that it was not my fault. I tried my best. If you were asked to do your work in a pressurized schedule, it is obvious that you will lose some of your qualities. So I felt, I felt that it was not my fault. It's their fault that we rushed. That's why the blame was not fair.

Researcher 19:01
So, now things have changed and improved the right people can come forward and talk about their mistakes right? Yes. So how this change of dynamic and improved your perception or your attitude toward quality? For example, do you learn from your mistakes? Do you share your mistakes with others? How this change in attitude from the management has improved things at work? Is admitting mistakes ok now?

Participant 5 19:35
Yes, I think it is very important to discuss the mistakes because your team members also get the chance to learn from you. And if they know that, they will try it. Try to avoid it. So I always keep open mind and share my mistakes with each other so that nobody repeats them and sometimes we'd like to discuss with our management so that they also understand the consequences. This openness has changed our thinking dramatically. We learn better and faster and it show in the quality of our work.

Researcher 20:09
So this openness that you created between the team and the management has improved things, right?

Participant 5 20:17
Yes, very much.

Researcher 20:19
So, did you notice any improvement in the quality of the code that you produce and the quality of your peers codes? And did you see any improvement in defects and things like that?

Participant 5 20:34

Yes, I see, I see the reflections and the improvements. And because when you are in a good mood, we are confident and less pressurized, we deliver better quality and less defects. So if we follow the right procedures and standards, the output gets better. Because it reduces the chances of mistakes.

Researcher 21:03
Great, nice to hear that I'm happy to hear that really good. Let's move to the next item. Which also was in the email which says member of your team can bring problems and tough issues? Your answer was yes, but not always most of the time, our team works under pressure. So they have to cover up task and the deadlines as deadline comes ahead. Sometimes teams have to overcome some of the problem and the issues. Before we discuss an example. Can you elaborate in this a little bit?

Participant 5 21:48
Yeah, well, there is a principle called YAGNI. In software engineering, this principle means you ain't gonna need then do not overthink it. So, for some time, sometimes we found this type of issue, one of my members come to me and say that look, this will be a problem when the picture will get bigger, but I thought I consider it or I think that I have a deadline in front of me. And if I try to implement the solution, we will miss that deadline. This application is now not big enough to face this problem. So sometimes we may overlook because, you know, we have to consider we have to consider the time also.

Researcher 22:54
So, can you share with me an example? Do you have an example?

Participant 5 23:02
Well, yes. Well, yes, let's say I have this project we are working with, like 1000 government offices. So there is a report system where we'll make report for individual government officer like in the 1000 offices, we have more than 160,000 users, where we have to make performance reports for each individual's. So, this is a pretty huge task. When we started the application, it was it was like it was we thought we should do reports for like 40 to 50,000 people, but it already reached 160,000 per slot, but my team member analysed the problem and he said look, it will not work for more than 200 people, but after that our server will be overloaded or the production demand on the server will be so high that we did not prepare for it. Because the process to generate the report start at 12am daily and usually takes four to five hours to generate the report. But if it takes eight to nine hours, it will even serious so, he came to me and show that to overcome that issue, we had to redesign the whole system. But it is very difficult to redesign a whole new system which is in production. So what we did we held a meeting with the stakeholders and we showed them that the limitations and we reported the problem. But we also said that after this threshold, you have to redesign the application.

Researcher 25:16

So how did the How did the management and the client reacted to this because this is quite a serious scalability issue. I think that the design failed to cater for the scalability. It wasn't communicated in the requirement upfront or what happened?

Participant 5 25:39
Actually, when we started the project, but the stakeholders also did not realize that it will be that might be so the demand on the generation of the reports is so high. So they had to consider it, but they didn't. Because, you know, it was very difficult job. I already mentioned there are there were 8000 government offices, each works their own individual way. Yeah. So we have to, if we want the system to work, we had to be adamant there in certain method that everybody follows. So it was a big challenge. That's why the stakeholders weren't certain that it will sustain. But when it already got, when it sustains it got a huge perfect performance. We got an award from the ICT ministry. ICT second runner up of ICT egovernment system.

Researcher 26:48
So let me ask some follow up questions with this example. So as a team, how did you talk about it? Did you feel it's okay to talk about this problem?

Participant 5 27:03
Yeah, I thought it was very important. Because after a certain time, we were certain that the system will not work. And we had to make a new version of it. The stakeholders were not aware of it. They will suffer, right?

Researcher 27:23
Yeah, of course. It's a failure. It's a big failure of the system which cannot handle the pressure. Yeah. So what did you learn from this mistake as a team or from this problem or other it's a problem? What did you learn from it? And how did you go about it afterward?

Participant 5 27:46
Well, I learned two things, one of them was forming and working with certain structures for software development, because when this project was started, it was not that well planned. So this project was poorly planned, but it wasn't. If it was, well planned, I think it could have scaled very well. And it could easily handle up to 1 million users. We also learned that we should not start developing without proper analysis, it may hamper the performance of your application. Now, we cater for scalability in every project we develop. Not only performance, but also a scalable design to accommodate future needs.

Researcher 28:33
That's great. Fantastic. So as a team you learned and you become better at what you do. That's fantastic. I'll move to the next items. I asked you in the email a question which is people on your team sometimes reject for being different. You said yes, this tendency mostly shows by senior member of the team to keep their influence sometime they reject different promising idea. That's interesting. Can you elaborate a little bit on this answer before we discuss an example.

Participant 5 29:14
Okay, which question was that? Yes, I'll put you in different Yeah, I remember that. When I first joined, I had I had to develop, I was working with the development lead. He was also an employee of that company. So the scenario was like that foreign project we did application, we needed a application front end application. So on that time, I read Java Script, really very popular for that because it has a very, very good performance. So, one of my team members suggested that we should use React JS to use develop this application, because it will be much more faster, better and good performance, etc., etc. But that team, a team lead team manager, sorry, the template who was also an MD, he rejected that idea and killed her idea. And his idea was that he was not very confident with React JS. Also, I thought he was unwilling to learn new technologies is for the for his project. So what he does, he rejected that idea and start developing with older technology contiguity. But the jQuery was much more slower. Development was much, much larger than React, because it is component base framework and jQuery was a library. So it took a long time to develop the application.

Researcher 31:12
So what were the consequences of this unwillingness to accept this initiative?

Participant 5 31:21
I think the consequences was the performance dropped significantly. And the development costs being much higher than anticipated. So I think, to see choosing the right person, for your project is very important. Those who make the decisions, they should be knowledgeable, and they should be willing to accept changes, ideas and initiatives. Because if you don't accept the changes, you go far behind. As you can see, we delivered again an application with poor performance.

Researcher 32:02
Yes. So you saw that the outcome was not in favor of the performance of the application? What was the atmosphere and the reaction to it? So did you talk to each other? Did you learn from this behavior, what happened?

Participant 5 32:28
Once the application was developed, there was nothing to do. A bit too late. But I felt that before starting the project, as I already mentioned, you should do proper research about it, weight design and architectural decisions and consult the team before you make committed decisions. And you shouldn't just go with a technology, because you know it and comfortable with it, you shouldn't do that. We should adopt, you should adopt technologies that add value and produce better quality. We should learn them, you should implement them and allow people to contribute. Because, we do not know everything. We all bring our own weight to the team. this will make the conditions of collaboration much better.

Researcher 33:08
Yes, so did the team embrace this openness and willingness to invest in research and learning upfront before commitment to these technological decisions that may impact the quality of the product?

Participant 5 33:26

Yes, this is interesting, because sometimes we use it as examples. We use it as examples in our meetings, and we tried to discuss in the team and we try and strive to avoid similar mistakes. We learned from this. Because it was it was a very, very good lessons for us. We remember it all the time.

Researcher 33:50
That's interesting, because it was it became a reference point for the team to go back to and to reflect on right.

Participant 5 33:59
Yes. Real life failures are powerful. You don't forget them.

Researcher 34:02
Yes, they are very powerful. Yeah, they make impact on the team, right?

Participant 5 34:07
Yes, yes.

Researcher 34:11
Okay, fantastic. That was a very good example. Thank you. I move to the next item on the email which I have sent and we can discuss it. It is safe to take risk and initiative in your team. Yes, we actually encourage initiative, but about the risk, we mostly discuss it. Before we go into an example, can you elaborate a little bit on this? So now, the dynamic on the team has changed and there is more openness and safety and people bring initiatives. Can you elaborate a little bit on this?

Participant 5 34:59
Yes. Yeah, now we try to encourage initiatives and before rejecting them, we do some research about it. Because good initiatives can change the outcome of project in healthy ways. I can also give an example. Yeah. Let's say, in our project, which we had to make a lot of PDFs for the staples. So we use the library called DOM PDF. It was working, but it was slow, and we use our own language to show that PDFs Unicode. So we use Unicode letters, when you will use the Unicode letters to show in this PDFs the performance degraded a lot. So it was getting a big issue for us, because we cannot generate or show those super properly. So one of our to one of our team members did some research. And they have like a very risky, but encouraging idea. So we use a new PDF engine, instead of library. It was a very risky decisions, because we make learning engine and maintaining a server was difficult. But in the end, we implemented a project called application to PDF and started implementing it, the performance boosted a lot. It was like a magic world where we are we were making and showing a report with two three minutes, it will take just three to four seconds to show that it wasn't really received but very successful.

Researcher 36:57
So let me follow up on this example. This is a great example. Thank you. What made you taking the risk? And following your colleague suggestion? What do you think motivates the team to take the risk?

Participant 5 37:15

Well, there are some factors, because we were desperate to seek another solution. And after, when he researched and implemented a small pilot to demonstrate to us, we found that yes, it's worth taking a risk, because our solution was on the verge of collapsing.

Researcher 37:43
So you did the integration. Yeah, that's great. Yeah. You didn't think of the consequences when taking the risk?

Participant 5 37:54
Well, yeah, we also discussed with the stakeholders, and we also took some risks if this method fails, how could we go back to our previous system? Like yes, so we took a risk with this method. So just because we also felt it is ok to take a risk and try. We didn’t feel there will be consequences. Yes, we were desperate for a better solution, we also felt it is safe to experiment. We didn’t want to deliver a poorly performing application.

Researcher 38:23
So this openness with your stakeholders encouraged you to take the risk and it did work.

Researcher 38:50
Yes. Okay. So in it somehow this openness and the buy-in from our stakeholders creates this safe feeling that we can take the risk and see what happens? So, as a team, this this became a behavior. So, what did you learn from it first, and I would follow up on another question, what did you learn from this from this process or this experience of taking risk, and it was rewarding to take the risk?

Participant 5 39:14
When actually we try to we try to analyse the pros and cons. Because it is not always easy to implement that solution in our systems. So we learned pros and cons and how we should make a decision whether it worth it or not? Because there is a bad side of openness to everybody wants to give their new ideas. So you have to calculate the risk before you decide to try it. Is it worth it worth it or not? What I mean is you have to be careful, we can not try every new idea. Some ideas are riskier than others.

Researcher 40:01
It is a calculated risk before you take the risk. That's what I understood. Yes. Okay. That's fantastic. We will move to the next one, which is the most important one, in my opinion, especially in a software engineering context, which is helping each other. The statement says it is difficult to ask other members of your team to help. You said no, we have very friendly team, we don't hesitate to help each other. So you help each other because you are confident that helping each other will make things better.

Participant 5 40:43
Well, in soccer, you know, there is a concept called Dr. B. Which is when you think about a certain thing for a long-term view by stacking certain parts. So discussing with each other, or showing it to other eyes can help there. And, you know, humans want to have their own capabilities. When you have your own limitations, some couple of hands may be very helpful for you. So we help each other because it's also

a way to share your knowledge with each other. Once you help your team member, he will get rich day by day, as they will be helpful to you in the future. It makes the team stronger.

Researcher 41:37
So it's a win situation. Right? You share? Yes. Yeah. So do you think this knowledge sharing helps also quality and how?

Participant 5 41:54
When, you know, software engineering, or coding is actually not inherited, I mean do don't learn is one day. So you don't build it in a day, you will grow day by day. So when I started my career, I wasn't very confident, or I wasn't very knowledgeable. And from my researcher, my colleagues, my teams I learned. So after certain time, I got confident that, yes, I think I am good at it. So helping each other will transfer this knowledge and make him a good make another person a good software engineer. So if I help my weak, weak colleague, the team will get better day by day. He will learn what to and how to do it. Coding styles for example, he will learn how to produce a clean code. This really help us achieve better quality.

Researcher 43:03
That's a great, it's again, a win situation, you become better at coding and your colleague become better at coding, and it's a win situation.

Participant 5 43:15
Yes. Exactly.

Researcher 43:16
So you have an example here, can you can you talk to me or share with me this example?

Participant 5 43:25
Okay, yes, there are two examples. Let's say let's say for the first example, that we were making a spreadsheet application, we have client. So there are certain challenges because it takes a lot of skill level scripting. It you have to know a lot some JavaScript to solve this case, console problems. And all the time, I was not really reaching that. So I started I was stuck in a certain situation and couldn't, couldn't overcome it. Then one of my team member who was who was quite fluent in JavaScript helped me he helped me to get to work on this situation. And also he helped me to know these procedures to implement their surgical structures. So together, we made a very good solution for that and better code quality.

Researcher 44:30

So do you think this collaboration made the code for the checks the spellchecker better? Do you think the quality of your code was better as a because you collaborated with your colleague who had bid who had more advanced schemes?

Participant 5 44:50
Yes, I learned from him. And also my code got much better, we made a very good solution, a scalable solution. The spellchecker was very influential.

Researcher 45:01
Okay, fantastic. It's nice to hear you have another example, right?

Participant 5 45:09
Yes. My other example was, you know, related to security of the browsers, you know, our application or web application, usually around some browsers. So after sometimes Google releases their affiliate their browsers to in 2018, we face a problem, it was not ideal. And in 2019, we faced these types of issues where Google updated their policies, and they blocked some of the some of the features due to security reasons. And our tradition was using those features. So after updating browser everywhere, everybody was getting some errors or pitch was not working properly. So and it was, it was quite an emergency to fix. So we, our team, we sat together and find the security vulnerabilities, because not all of our supplied certificates were expired. Not all of us has expired security certificates. So those have knowledge they shared with us and we fixed them. So we collaborate with each other and work on the fixes together. The problems were fixed faster and with optimal solutions.

Researcher 46:31
Okay, that's another great example. So this collaborative work and solving this security issue. Do you think it happened because people feel comfortable in their work environment to help each other to solve these software issue?

Participant 5 46:57
Yeah, it's the environmental. As I said, we feel comfortable and safe. And we feel it’s ok and happy confident to take initiatives.

Researcher 47:06
So because it's a shared belief, you share these values together, and you feel it's safe to do so. Right?

Participant 5 47:14
Yes. We believe that we need to help each other.

Researcher 47:17
So what would be the consequences? Yeah, so no problem. So what would be the consequences? If for this, Google security update policy, for example? If you didn't call up alright. And you didn't help each other? What would have happened?

Participant 5 47:42

Well, our application was gonna stop working. So we would have been at risk to crush the client application in production with some business consequences. Because if we talk to our stakeholders, they would actually ask us why you had vulnerabilities, or why did you make a decision that has security issues? Right. But that time, it wasn't an issue. After sometimes it becomes a policy and a Google requirement. But we didn't anticipate that. So if we did not overcome the problem, it would have been a disaster.

Researcher 48:21
Yeah. I can imagine. Thanks. Thanks for that example. It was a very good example. The next item, no one on my team would deliberately act in a way that undermined my effort. No respect, your answer was no respect each other. And we try not to undermine each others. And I'd like to make reference which in the early example, was your project manager stepped in to support you? How did you feel when he did that? Or she did that?

Participant 5 49:01
When I felt relieved, because I was very frustrated, I was angry. But when he stepped up, he supported me, defended me, he talked about it he talked to the management, and he also explained me the things that may have made me felt better.

Researcher 49:22
So did your motivation picked up after he did that?

Participant 5 49:29
Yeah, because after that, after the blame, I was very, very frustrated. And at that point, I was not willing to work for them. But after he discussed with me, then I got motivated. So I think yeah, okay, it's fine. It is also a lesson for me to not to rush things.

Researcher 49:55
Okay. So it's not only motivated you. Was it also an opportunity for you to learn because he supported you?

Participant 5 50:06
Yes. I learned and become more confident. I knew I had the right support. I was really good.

Researcher 50:08
Okay, great. We get into an end, which is the last statement. And I do have more questions after that. So working with members of my team, my unique skills and talents are valued and utilize. Unfortunately, you answered no, not always, sometimes idea or different method may get rejected because of being Junior. Can you elaborate a little bit on this? I think we already discussed this before that in some cases, yes.

Participant 5 50:44
Yes, yes. When I joined the team, that I faced this situation, but I just better now, which is better now.

Researcher 50:55
Do you think that junior member of the team still faced the same attitude?

Participant 5 51:02
No, not anymore. Okay. Now we actually discuss it.

Researcher 51:06
Okay, great. Do you have another example? Can you share that with me?

Participant 5 51:20
Well, it is also a situation used in that time. Okay. Because when I told you, we had a lot of users in our application, so you know, when we had a lot of users, we struggled to manage our sessions. Sessions means like a security token that handles the users. So we had a big, big users big set of users, and we were struggling to manage it. So I started looking at a solution called Redis which is a session management server. But I was new. And I think that idea was well, because I was so young to service the state of big changes. But good news. Now we use this in our applications. We implemented it last year.

Researcher 52:23
Okay, great. Well, how did you feel about that? Did it motivate you to bring initiative like that forward? After that?

Participant 5 52:32
Yes, I got very much. It motivated me because it's working now. It's working. It's a very good solution. Most people, many people using it. I think, I was suggesting with that idea on that time, it was a very brave idea. That was overlooked, but I'm happy now it's working. And I learned that I should not reject anybody without preparing properly or analyzing the idea.

Researcher 53:12
Great value. Yeah, you become more open to other people's idea, right?

Participant 5 53:17
Yes, very much.

Researcher 53:22
So in this example of implementing Redis for handling sessions, how did it help quality?

Participant 5 53:34
Well, if you have a lot of servers, a lot of sessions in your servers and your server is not efficient to manage it, the applications performance gets slower. But when you implement rival services, separate server, so it did not affect any performance of your applications. So you, can you eventually get in better performance. And as you said in the ISO definition this is an important aspect of quality.

Researcher 54:00
Okay, fantastic. That's wonderful. That's a very good example. Thank you. Thanks, Participant 5 I think we came to an end. But I'd like to ask the last question. Is there anything around this environment of feeling safe feeling at ease to speak up bringing initiative? Feeling confident to take the risk calculated risk sometimes in your team? Would you like to share anything that we haven't discussed?

Participant 5 54:37
No, I think we have covered pretty much all but one thing I have one bad experience to share with you that our MD, who was also an employee in our company, I had I have bad experiences with her since she is an MD. He tried to over forces us.

Researcher 55:04
Director or a managing director? Yeah, management. Sorry, thank you continue.

Participant 5 55:15
Since she is a manager and direct managing director, she always try to overpower us. And then I did not like this idea. Because when we are to when we express our interests or express new ideas, how we usually do some research for us into some research, who we think is the pro and cons, if we think yes, this is a bit better solutions, then we speak to the team with the team. But after that much hard work, if your ideas got rejected, or if he overpower you, then I think it is a very bad behavior. Eventually, she went to other teams, but I said that an employee shouldn't be an employee, if she got involved with a company, she always tried to take the opportunities, or misuse her powers sometimes.

Researcher 56:27
So in this example, how did you feel motivation wise and taking initiative? How did it affect you?

Participant 5 56:37
I think managing director should be in the management, she shouldn't be involved directly with the team. Because if he misses his point, have nothing to do. He cannot talk with others about our management people, because they're already in support of him.

Researcher 56:56
So that brings me to an important question, how self manage Do you feel? Do you feel in your team that the management interferes a lot in your work, or you have some level of freedom to make your own decisions?

Participant 5 57:17
Well, it's a balanced situation, because you because if I think feel to take some necessary steps, I have to discuss with other team members. So I discuss with my team, and I also try to explain to the management to justify my idea. We usually make our own decisions but sometimes we need to consult the management.

Researcher 57:49
So it's an open relationship, right? It's, there is no excessive control, but it is an open relationship. Everybody is talking to everybody, right? Yes. Thanks, Participant 5. I don't have more questions unless you have some questions for me.

Participant 5 58:16
No, I don't have any other questions for you. Okay, that's a pleasure to talk with you.

Researcher 58:21
Yeah, it was great. Thank you. I will stay in touch. Thank you. Thanks for your time. I enjoyed the discussion.

Participant 5 58:30
I enjoyed the discussion so much.

Researcher 58:32
Okay. Bye bye. Yeah, yeah, I will. I will. I would get in touch if I need you. Thank you very much.

Participant 5 58:43
Okay, thank you very much. Bye. Bye. Bye.