

Participant 16

SUMMARY KEYWORDS

team, developers, tests, testing, problem, qa, quality, discussion, question, work, fantastic, mistakes, discuss, product, meetings, case, test cases, release, answer, effectively

SPEAKERS

Participant 16, Researcher

Researcher 00:25

Hello. Hi, Participant 16. How are you?

Participant 16 00:31

I'm fine. Thank you. How about you?

Researcher 00:33

I'm doing well. I don't know, what's the problem with my camera? Just give me a second. Yeah, sure. For some reason it's not working, but we don't need the camera. If it works later, I can put it on. So thanks again for accepting to do the interview. I do really appreciate, I'm just gonna explain to you how we're gonna proceed with the interview. And then we can start afterwards. Sure, Thank you also for the interview.

Participant 16 01:24

Happy to help.

Researcher 01:26

Okay, The interview is for a research study. So I'm doing a research study to see how agile teams work in order to achieve better software quality. And in particular, I'm more interested to see when they admit mistakes when they bring problem forward, and when they take initiative and how much it helps to improve software quality. So I do have questions, we will go through these questions that it is quite fluid and flexible at any stage. If you'd like to add something or if you want to discuss something related to the topic, please feel free to jump in and provide your input or comment. So it's not a rigid by all means it's a discussion.

Participant 16 02:32

Understood. All clear. Yeah. Let me please add that the scope of this research is, is quite an interesting one.

Researcher 02:40

Yeah. Thank you. I'm glad you find it this way. If you don't have any question, perhaps we can start with a brief presentation of yourself.

Participant 16 02:51

Sure, let's start with that. So I'm working as a QA quality assurance for like eight years, but 15 years in software development. Now, I have started of course, with manual testing, while I was doing consultancy work for a company that was producing an ERP software. And I was a consultant I was doing also the sit the marketing side of the of the project and also effective implementation and afterwards taking care of the customer problems and questions and so like support. So, in that point, I was doing mostly manual testing. And then I moved on and I realised that I really liked the testing part of all these projects. And I started specialising into testing, I then moved to another company, which was producing a software that was used to manage the industrial, the industrial sensors, used in factories, for example, on production slide on production lines. And that was the first point in which I have done automation testing. Afterwards, of course, I followed this path. And I started going deeper and deeper into automation testing. And in my current position as a senior quality assurance, I'm doing exactly this. So my daily, more or less my daily actions presume checking manually the new features that are launched on our website and also assuring the quality through different executions of the tests that we have beat smoke tests, regression tests, sanity test, either on the testing environment, staging environment or effectively production. So this is shortly said, my background.

Researcher 04:54

Okay, great, thank you. That was precise and that was sufficient. Thank you very much. We will ask you questions about your team and the type of software and we proceed to the main aspect of the interview. So are you using Scrum or other forms of Agile?

Participant 16 05:17

Yes, we are working in Agile Scrum. But at the moment, we have switched a little bit to Kanban.

Researcher 05:24

Okay, can you take me through the process how you use Scrum or Kanban? And briefly just explain why you use why you switch to Kanban?

Participant 16 05:35

Yes, sure. So we switched to Kanban for a simple reason. We are around 13 developers in our team and two QAs. And due to the fact that we want to approach multiple projects in the same time, this team of 13 developers will be split into squads. So form small groups of three, four developers that can approach a project starting from the beginning until the end. And therefore if we would work in Scrum, and have all those meetings and a lot of discussions Well, us from the key point of view, would be overwhelmed with the number of meetings and we won't be able to effectively do anything. And that's why we switched to Kanban. Therefore, we asked QAs are able to follow the projects, the parallel projects and also contribute effectively with the with K work.

Researcher 06:29

Okay, fantastic. You mentioned you are cross functional teams. Do you have other functions within the team other than the QA and the developers?

Participant 16 06:42

Well, yes, fortunately, or unfortunately, yes, we also do. We also handled the release for our process. We also take care of some documentation regarding the quality assurance mainly. And, yeah, I also, from time to time, we are maintaining some of the test pipelines that are getting triggered in Jenkins. So in our continuous integration,

Researcher 07:11

Okay, great. So, what's the size of the team? You said? 13? Plus two?

Participant 16 07:18

Yes, 13 Developers plus two projects.

Researcher 07:22

Okay. How long have you been working together? Is it a new team or you have been working together for a while? And would you consider your team self-organized?

Participant 16 07:28

Well, I would guess this is my fourth year in this team. So we have been working for a little bit. I would say to a great degree, we are self-organized.

Researcher 07:34

Okay. The last question about the team. What type of project do you developed? And it is a new development or enhancing an existing one?

Participant 16 07:40

It's a web application. Both, many functionality are new and we enhancing existing ones.

Researcher 07:42

Okay, fantastic. Okay. Let's move to the to the main aspect of the interview because the study is related to software quality. I think we need at the front discuss what do we mean by software quality. And I'll get your perspective and the definition we use. We use we use ISO standard definition, I read it to you and then we can discuss briefly. So the definition says software quality is the degree to which the system satisfies the stated and implied needs of its various stakeholders and thus provide values. So the model also propose some non functional characteristic like performance, compatibility, usability, reliability, security, maintainability, and portability. First, do you agree? Or would you disagree with this definition?

Participant 16 08:42

Yes, I agree with both. And we are trying to do our best regarding both of the options. Okay, so expressing usability opinions from the user's point of view, and also trying to get along with the business requirements that needs to be met within our features.

Researcher 09:01

Okay, fantastic. So in order to achieve quality, you discuss some type of testing, smoke testing, performance testing, what other quality assurance processes and practices you have in place to achieve this quality?

Participant 16 09:19

Well, we are not doing too much of load testing, stress testing and performance testing. But more we are doing like smoke tests on the end to end tests. So it's more or less on the user interface level. And here we apply mostly blackbox testing. Of course, we also have some, integration tests at the API level, which is a middleware between the front end and what kind of said middleware between the front end and the back end and even between the back end and the DB.

Researcher 09:54

Okay, great, fantastic. So the developers, I would assume they do some code reviews, right?

Participant 16 10:01

In most of the cases, yes, they were open with pull requests. They work with pull request, and they give code review.

Researcher 10:09

Okay, great. Thank you. Thanks for sending those answers on the email. I appreciate your work. One of we made an assessment of your team. And we do think based on this answers, your team is highly a safe work environment. I will explain what do we mean by safe in this context, and I get your, your comments. So we mean that when we say safe, we mean that the work environment provide a sense of security from repercussions. So people in your team feel that it's okay to admit mistake, they feel that it's okay to propose initiatives and discuss problems properly. So within the team, there is a good sense of confidence that the team will not embarrass or reject or punish someone for speaking up. So there is a confidence in the team that they won't be respected, they will be trusted on the team. So we do think that is highly safe with this definition. Would you in a scale of five, do you strongly agree? Do you agree? Do you some what? Neutral? Or these agree or strongly disagree?

Participant 16 11:37

I totally agree. And you said on a scale of one to five, right?

Researcher 11:42

Yes from strongly disagree to strongly agree.

Participant 16 11:45

Yeah, I strongly agree with what you said about my team.

Researcher 11:49

Okay, fantastic. So what's made the team like, safe? What do you do or what happens around you on the organisation to make this quality people feel safe, they admit mistake, they bring problem forward, etcetera?

Participant 16 12:13

Well, I will try to more or less sympathise toward each other's in the entire process that we have here. And I would like to believe that we ask you, as were the drivers of this change. So a few years back, we were also working, not the chaotic but we didn't have all these rules and all this trust and respect in place. So therefore, we have more or less the took some responsibility on our side, like developing testing pipelines, like portability in developing some quality assurance gates from whenever the tickets are passing from the developer to the QA. And we also said that we want to do automation, because this will actually help the entire team. So more or less, we started some unpleasant discussions with some developers, because you can imagine this implies some change in their way of working, starting from how they create the pull requests, the strategy of the pull request, merge, and so on, I won't enter into details. But actually, we have started those discussions because we wanted to make our life simpler. And since we weren't the ones responsible for doing the releases, we wanted to make sure that we have a high level of trust, have confidence when we do that, that release. So therefore, we have introduced the different suggestions to the team. At the beginning, they haven't really liked them. I mean, you can imagine nobody likes somebody to just come and suggest different changes to your work. But in the end, they realise the value of those changes. And yeah, that's how we got into this position. Okay, fantastic.

Researcher 14:01

So it was a team effort, something that it has emerged from the team by being open and communicating and deciding on our way forward, right.

Participant 16 14:13

Absolutely. I mean, we couldn't have done this without, without the support of the team, we would have suggested 1 million things, but if the team in the end wouldn't have agreed them, yeah, we would have done nothing.

Researcher 14:24

Okay, fantastic. Thank you. Now what we're gonna do, we're gonna go through the answering your providers on the email, and we're gonna discuss one by one. And hopefully we can also discuss some examples. So the first one is, if you're if you make mistakes on your team, it is often held against you. Yes, say no, nobody will hold anything against me. Everybody understand that? We are all people and people do mistakes. So what does happen in the team when somebody breaks? Because a mistake forward and say this is what happened. This is what's happening? And what how does the team engage and deal with the situation?

Participant 16 15:12

Well, I would, I would take the latest example that happened during the last days. So for example, we have done some release on the live environment, right. And there was a mistake, and our users could not access the products that we are selling more or less. And we have realised that it wasn't from the developer side, it wasn't any mistake from the QA side. But it was misunderstood the requirements from the data team. I mean, they have modified some information that went into the live database. And that case was not more or less treated. And therefore, any result list regarding anybody's was not available. What we did, we actually first tried to find the root cause to find the root cause, then we found it, and then we actually more or less did a roll back in order to actually bring again, the live environment to the desired state. I mean, working.

Researcher 16:20

So that's quite was quite a serious issue, wasn't it? I mean, it was a big issue. Yeah. So how did you dealt with it? I mean, you did a root analysis to find the problem. But from a social perspective, how did you deal with it as a team?

Participant 16 16:45

Well, we communicated to the data team, we should more or less keep the communication tighter, because it brings like a negative image upon the entire team, not them, or QA or developers. I mean, we like to think that we are one not doing any differences between the different parts of the team. So yeah, we try we are trying to tighten up the communication in order to kind of avoid these kinds of mistakes.

Researcher 17:17

Okay, so when you say avoid this kind of mistake, do you mean that you learn from this type of mistakes?

Participant 16 17:24

Hey, we always learn from the mistakes and those are the most valuable learnings that you can have.

Researcher 17:32

And how do you? How do you? How do they influence your decision in the future?

Participant 16 17:41

When probably, probably, we would, we would like to extend the test cases that we have in order to try and actually also cover the problems that have occurred, and we're not covered with tests. But of course, that won't be possible. I mean, we are expecting more issues to arise more issues like that in the near future. But we're doing our best if we can avoid them fine. If not, well, anyways, we will do our best.

Researcher 18:13

Okay, fantastic. Thank you very much. Do you think when people admit these type of mistakes and bring them forward? Do you think it influenced the quality of your product? Somehow?

Participant 16 18:29

Well, probably from a user point of view, yes. That's something that nobody desires. I mean, just accessing the website, then you find the blocker and yeah, but otherwise? As a development team, I mean, we learn how to do things better. We learn our weaknesses and how we can fix them.

Researcher 18:57

Yeah. So what do you say in here? Sorry.

Participant 16 19:08

Yeah, so it brings a negative influence on to our product. And of course, yeah. Bad image to the product itself.

Researcher 19:18

Yeah. But from an outcome perspective, you said you learn and you avoid, hopefully the mistake in the future. So there is there is some impact on the team isn't?

Participant 16 19:35

Well, on the team itself, no, but on the business on the company business, yes. Because whenever this happens, we are losing money.

Researcher 19:43

Yes. Yes, of course. But as a team you learn as you go from these type of mistakes, right?

Participant 16 19:45

Yes, definitely.

Researcher 19:48

Yeah. Okay. Thank you. I move to the next question, which is not Number of your team can bring up problem and tough issues. And your answer was yes, every man that can do this, we actually empowers the importance of the open discussion. If there are something that the team can help with, it's great. Can you take me through what happens when people bring up problems?

Participant 16 20:27

Yes, for example, so somebody says, Hey, I'm in factoring this discount here. And I would like to do it this way. This would imply that there are implications on to this side of the product and that side, what should we do? And we from the key point of view, we are saying we are having discovered through these tests and these tests, the business side says, Hey, this would be important because it will affect whatever the users would see, and bring on their, their poor ideas regarding that, and so on. So we are still doing that change. But we are just making sure that this won't, won't 39 Or bad influence on our website.

Researcher 21:13

So you talk about open discussion. Can you share with me an example of an open discussion around that problem?

Participant 16 21:24

Yes, for example, the code base of some parts of the application is pretty old, it has several years, since he's there, and some developers are willing to take this up and change it. And therefore, whenever they have something that we should also know, they are sending out a meeting invite, they're inviting the POS are inviting us the QAs and also some other developers that that are involved in know that part of the code. And then it starts a constructive discussion. They're telling us what they want to do, how they want to change it, what would be the benefit, the benefit, the disadvantage, and the touch points that we should take care of?

Researcher 22:10

Great, what how does it make you feel you as a QA and other QA, when you are you get engaged and involved in this process?

Participant 16 22:21

Well, it says normal, right? I mean, that's how you should actually be. Since we are all a team, and the entire responsibility of the product comes down to us, we should take decisions in a joint meeting to actually still keep the quality high. The product working.

Researcher 22:43

Yeah. Yeah, sorry to interrupt. So when, when you feel when you get involved in this level of from the start when the initiative is taken by the developer to rework the old code and you get engaged and unfold from the beginning? Do you think it helps your testing or your approach to improve the quality of that part of the code?

Participant 16 23:09

Well, probably not at the beginning, because you can imagine there are parts of the discussions that I don't understand the technical parts of discussions. But still, I have to get the big picture. So more or less, they have to translate to us POs and QAs is what they are doing and how this will change the product. Because we have to be aware of that and contribute with the with our, with our work, more or less to test great more tests or modify some of the existing ones and the bills to just express their opinion from the business side.

Researcher 23:47

Yeah, so you become knowledgeable what's gonna happen, and also it gives you an opportunity to influence or to add your perspective, because QA and most the time they are more knowledgeable about the product from a user perspective than the developers.

Participant 16 24:03

Exactly. Even if it's like a purely technical change. We are still asking, Okay, how can we test that? Even if there is no chance that that we would be able to test that? Still, it would be comfortable to just know, hey, you can do this as a manual check. And it should be fine.

Researcher 24:26

So this conversation helps both party you and the developers. Yes, it's brilliant. Yeah, it's brings you in line so you understand the impact on the quality on testing. And it's also helped them to acquire some knowledge about the product.

Participant 16 24:46

Exactly. Yeah.

Researcher 24:48

Okay, thanks for that. Good example, I move to the next question which states people on your team sometimes reject other for being difference? And your answer was not really our team has mentored from all over the world and have been working like this for a long time. So what we meant here by a rejection is not based on nationality or ethnicity. It's nice to know that you are diverse team. But what we meant here is the rejection on base of your style and your opinion as a team member. How do you like your question is now because you sound like I'm very open team. So imagine I was a member of my team of your team, and my approach is a little bit different than the rest. So would they be rejected? Because I do things differently?

Participant 16 25:53

Well, not really. So in our team, the majority of the team has a high seniority level. And more or less, if we talk about the product, then we will understand what you would like to do even if you say it nicely, or even if you say, like, bluntly, it's, it's fine. In the end, what matters is the idea that you want to express and what you'd like to do. I mean, we're all different things. Maybe some have some soft skills and can express their opinions in a nicely and elevated manner. But some others are developers, they shouldn't be like I know, businessman, or marketing people and express everything as nicely as possible. Now, what matters is the end what you want to do and how you want to do it. So it matters for us to understand what you want to do.

Researcher 26:45

So the outcome matters more than these.

Participant 16 26:49

Yes, exactly.

Researcher 26:52

Thank you very much. I move to the next question. It is safe to take risk and initiative on your team. And your answer was, depending on the impact, in my case, as a QA, we have the freedom of choosing how and with what would to work, if this doesn't have any negative influence towards the team activities. Can you elaborate a little bit further in this question this answer?

Participant 16 27:24

Yes, for example, the question is, it is safe to take big risk initiative on your team, we have recently decided that we would like to change the testing tool that we use, and migrate all the tests that we have on to a newer one. Of course, it has advantages disadvantages, that would mean that the quality in our team would be a little bit overwhelmed for the period in which the migration is happening. But still, the team accepted our proposal and said, If you think you can go with it at this won't lower the QA quality that you are doing that is totally fine for us. So more or less, they trusted our decision. They are respecting our decision. And they will help whenever we are requesting help, because they understand more or less to aid in this case going up in this case, gaming is also part of the development. So we also have to update them with tools and the tech stack.

Researcher 28:34

Okay, fantastic. Do you think that this healthy and constructive approach and welcoming initiatives helps advance the quality of your product?

Participant 16 28:48

Well, definitely, I mean, in the end, that's what we also follow. Because only just updating of for the sake of updating, no, that's not what we want, we actually want a tool that will allow us to be more flexible, create more types of tests, and in in an easier manner.

Researcher 29:08

It sounds from the example you shared with me that you engage in, in a collective decision making you discussed together you inform each other before you make a decision to move ahead with an initiative or not correct?

Participant 16 29:25

Yes, that that's the case in in the majority of discussions. I mean, if there's something related to the release and how we document the release, hey, we won't do it in gyro or clickup or whatever. And it's a small thing then we will do it because in the end we are using that process and what matters is the final result, how the team will be able to track the releases. But something like that to that will impact the entire team. Of course we will do it in in a common meeting.

Researcher 29:58

Fantastic. Thanks for sharing Those examples. I've moved to the next one, which about helping each other. So obviously, in Agile team is about collaboration and helping each other. So the statement says it is difficult to ask other member of your team to help. And you say no, all our members are eager to help when they are approached and asked to help. Can you Can we start a little bit what makes this atmosphere of people willing to help each other?

Participant 16 30:41

Well, let's go down to effective example. So for example, we have a new feature that needs to go out on the website. Right. And from the documentation point of view, we could always do better. And some things are not enough document. We as creators, we are not interested too much about the technical

way in which the developers are implementing stuff. But we are interested about the user's perspective of the outcome. So for example, we often go to the developers and ask them, Hey, I would like to create some tests here. Can you please tell me how I should approach the tests and what Gleeks? And what options should I take in order to get a better coverage of your code? So this is one example in which the developers have implemented the stuff and will help us to more or less build better test cases.

Researcher 31:41

Okay, fantastic. So obviously, this keep advancing the quality of your of your software, because in this examples, you improve the coverage of your test cases, right?

Participant 16 31:56

Yes, of course.

Researcher 32:00

Okay, thanks a lot. I move to the next example. Do you think that this work environment where you work in helps also the relationship between the developers and the QAs?

Participant 16 32:19

Well, when the relation is based on trust, then it would surely help. Because each question and each detail that actually got a better ventilation is helping both sides. But if it's only done to I know, do your job and, you know, don't put a little bit of love. In the end, it's about responsibilities and respect.

Researcher 32:50

So this this in your team, do you think this responsibility and feeling accountable, accountable for your work and respecting each other? happening?

Participant 16 33:03

Yes, definitely.

Researcher 33:05

How does it help, the work and the task and the communication between you and the developers?

Participant 16 33:13

Well, for example, it doesn't make meetings for too boring, because we can discuss about a new feature inside the meeting, we can discuss it at the general level, from the users perspective without bringing any technical sides into the discussion. Because we know that in the end, we will also get there because we as case need to testify them as developers need to implement it. And one way or another, we will align on that. So we don't need to make the meetings too, too long meetings.

Researcher 33:54

So it makes you more efficient and the communication flow more fluently, right?

Participant 16 34:04

Yes, that's my experience.

Researcher 34:09

Thanks a lot. I move to the next items on the question. No one on my team would deliberately act in a way that undermine my efforts. And you say in most of the cases, the answer is yes. There are a few exceptions. Like for example, we have a Jenkin cluster, in order to execute automated tests, the tests are taken into account in all development pipeline, but some developer took the test out. Sorry, this is quite interesting, because it does happen yes. Due to their flackiness generated by the testing environment and stability. That is Uh,

Participant 16 35:01

Yeah, that's life. I mean, you can't have the demo, you know, I mean, we have decided, as a team that we want to create automated tests. After we had those tests, we have explained what why the tests are something is failing, effectively failing. And that is because the test environment, which is not handled by us, it's handled by the DevOps, we have a debate, whatever it is. And due to that test environment, which sometimes it's working badly, the tests are timing out, which in the end generates a failure is that developers are constantly working, and they're pushing code into the master branch on to the testing environment. I mean, I understand that this creates some kind of stress, whenever they're doing the push, the tests are failing. They're thinking what, hey, I did something wrong, and they need to actually lose time to see that in the end is not an issue on their side, but more like an infrastructure issue. Okay, this, I understand, but the fact that they took some tests out, and they are not relying on the test anymore, this I didn't get. So this was one case, and it happened only once, like in in the 40 in the four years since I've been here. But still, it's an example. The other case is more or less they are contacting us, they are saying, hey, the tests are failing, would you please take a look and stuff like that? But sometimes they are, as in this case, they took them out because they are failing too often.

Researcher 36:39

And why did they think they took them off? Because it annoys them?

Participant 16 36:46

Or does because they are annoying? They, the tests are generating all kinds of warnings in our team chat in our on our email. I mean, they, they don't care, because effectively what they did is it's fun. But really, they took the test out they cannot be right. Yeah, anyways, they will have to take them into account when we are doing the release. So more or less, it means a little bit more workload on our side and

Researcher 37:18

just the lie in there just delay in dealing with the with the problem. Sooner or later, it will it will catch up with them anyway. Yeah.

Participant 16 37:30

Yes, it has to be like that.

Researcher 37:33

That was a good example. Thank you. But overall, you don't feel like the team deliberately undermine your efforts. In general?

Participant 16 37:35

No, definitely no.

Researcher 37:37

Yeah. Okay, thanks. We come to the last item, and we will discuss it and we will wrap up after that. So working with my team, my unique skills and talents are value and utilise. You said yes, my team relies on my testing expertise in order to get the job done. How does it make you feel when you're when you're skilled or acknowledge in your team?

Participant 16 38:16

Well, of course, it gives a comfortable feeling, because you're taken into account because your work is getting appreciated, and you are recognised as a professional. So that would be the first but then it's also another person feeling the fact that you could actually help with the work that you are doing. And therefore it's a it's a mutual nice feeling.

Researcher 38:42

So it makes you feel like you're influencing you. You're contributing, right?

Participant 16 38:48

Yes, in the end. That's why we are here to bring our contribution.

Researcher 38:55

That was all the questions. I think you were very efficient and providing good examples. Thanks a lot. Before we conclude, is there any things you'd like to add? In this topic, which I didn't mention, or you like to add something we didn't discuss, you find it's irrelevant, or any things you'd like to further discuss in this topic?

Participant 16 39:25

Well, I'm not really sure. I mean, we have went through a lot of discussion. So I would guess I would guess it's a complete set.

Researcher 39:36

Yes, yes. We went through a lot. Yeah. Yeah.

Participant 16 39:40

By the way, once your study is finished, can I also get the link of your results or?

Researcher 39:47

Yes, it would take a long time before it's published. But what I can do, I can send you the draft before we send it for publication. If you like I can send you Did after the paper Yes, that's not a problem.

Participant 16 40:03

Yeah, that, would be great. Yeah. I do. I mean, I'm not effectively interested about my answers because

Researcher 40:10

No, no, no, actually we don't. We don't we don't. We don't show the answers of everyone. Your participation, as I said, is anonymous. We may use quotes from your answers, just that we use quotes from your answer from somebody else answered. Yeah, but the paper will be a synthesis of the outcome more than the answers. Yeah. So it won't be boring. Reading the interviews now. It has a lot of analysis on it.

Participant 16 40:47

Great. I would be really grateful if I can receive some

Researcher 40:51

Yeah, yeah, no problem. I'll get in touch and please stay in touch. And thanks again for your good example and the time to do the interview. I appreciate your work on okay. I wish you a good afternoon. Bye.

Participant 16 41:07

The same to you. Bye