

**P07**

**Interviewer** 0:01

Good afternoon, so with your permission, I'm recording the audio now. Let me introduce myself first. I am a PhD student at Federal University of Bahia, here in Brazil. I started my PhD last year, and one of the big areas that we are investigating in my study, it's in the intersection between software quality and neuroscience in software engineering. So, one of the things that we are interested in is understanding the QAs head, the brain area that is activated due to a specific QA task, for example, but this will be part of an experiment that we will perform later. But this it's a qualitative study, with interviews and then we will perform a survey with more people to validate the findings from this study. And for this one, what we are looking for is that we want to identify a set of attributes of QA in the sense of what are the main attributes that makes this person be a great QA, you know? So, the overall idea is for you to think of someone that you think it's great QA. What does this person have that you think that she/he is great at his job? So that's kind of what we're looking for... So, to start, I would like to know a little bit about you and what you do in your job as QA, your role, what are your responsibilities? Can you tell me a little bit about yourself?

**P07** 2:29

I kind of have two hats, like all QA team leads that are at <company>, we not only are team leads for managing people, but we also have the opportunity to continue being QA engineers, in our teams. And in my particular team, we work on the, what is called the Phoenix or portal scrum team. In that team, my responsibilities are to work with the product owner, the scrum master and the developers, and the other QA on my team to develop acceptance criteria for stories of which the developers will develop, and we as QA will have an opportunity to test. It's important to be a part of that development and be a part of creating the acceptance criteria to match customer desires, and in this case, the product owner will want from this JIRA story. And it's important as a QA to be involved in the development for whatever it may need, because it gives us the opportunity to understand the feature as a whole, but it really is meant to be and what it and by conversely, what it is not meant to do so that gives us a good way to look at what we were developing and being able to test it in ways that it should be doing running. But also, in ways that it should not and make sure that it never will run that way that should not be running. Okay, so that is my area.

**Interviewer** 4:44

ok, if you have a management role as well, in your scrum team, do you also act as the mentor of the QAs in your scrum team or do you actually work as a QA as the others QA?

**P07** 5:05

Yeah, given my position as QA team lead, it tends to put me in a position where they look at me as a mentor, and so on. And because I have a large amount of experience

compared to them, I can take that time to mentor those who are in our scrum team and give them kind of heads up and be able to stand back and let them do some trial and error. And just make out some pointers. And if they need help, allow them to use me as a resource to give ideas.

**Interviewer** 5:54

The first attribute that you mentioned on the form, I mean the first thing that was prioritized as the most important attribute was the desire to learn. Can you tell me a little bit about this?

**P07** 6:12

Sure. In some positions, it's not all that important. There's only so much you can learn. And that's all you need to know. And I'm talking about certain jobs. But when you're a QA engineer, for me, the most important part is always being willing to learn. Because there's always something new that is being developed, something new that is being tested, a new way to do something, cybersecurity, quite frankly, I see hackers as bad actors that are QA minded, because they are always looking for new ways to do something. Obviously, and typically, it's new ways to use the way that it should be. But it does require that you're always learning. Yeah, for me, always learning, the desire to always keep yourself open to new ideas,

**Interviewer** 7:31

Do you have a self-experience or from someone that was great at his job and fit this scenario in the sense of keeping learning and finding new ways to do something, and this was helpful for the project somehow or even the process.

**P07** 7:55

Yes, and this is both my experience and another friend of mine's experiences, as QA engineer in McAfee, my prior jobs. Both of us have taken the opportunity to continue to school. He has since graduated with his master's in cybersecurity. And I've done some certificates for cyber security through a college here in Idaho. But in doing so, I have found a lot of what I learned really worked well in my QA positions to be able to test and think of using something in a different way just to verify that it cannot be used in a way that it is trying to use it. and it's particularly important in the web in the way that we use our web applications in the way they continue to evolve. So, for me that, that learning helped me understand how people could use the web application.

**Interviewer** 9:29

When you mentioned "desire to learn", would that be related to learn everyday more about the business rules of the company? would that fit in this particular topic as well?

**P07** 9:49

Absolutely. So not only learning the technical aspects, but the business aspects of it because, really, that's why you're there and why your company is there is for business

aspects. Even if you're talking about the foundation for providing services to people, it's important to understand what they do as a business, whether they're charitable business, or governmental business, in my mind, they're all a business and they're all providing a product for someone. If you don't understand what they're trying to provide, it is harder to test out something how it should work and how it should not.

**Interviewer** 10:53

Okay, the second one was to question the known, what would that be?

**P07** 11:03

It would be if somebody is always doing something, because it's just always been done. That is something to question. It's something important to look at and not just accept as this is how it should do... When we do that, we evolve ourselves and we're doing the first thing I said that is always wanting to learn. I spent a lot of years prior to my QA role, as accounting, and a lot of tests in accounting..., because there are a lot of rules in accounting and even those that are set can be questioned and should be but what you find is that when you don't question it you get some serious consequences. In our current business, we are required to follow Sarbanes Oxley, in the US it's a law that Businesses have to provide in the financial space, particularly. And those laws came about because nobody questioned a company called Enron nor did they question financial statements that were certified by another large accounting firm. They just accepted them all and by doing that they didn't recognize that accounting firm And Enron The company has been hiding what they call cooking the books, if you understand that statement. They were cheating and nobody was there to question any things that just didn't seem right. Both the company itself and the auditor, that should have said Hey, it is a problem. We're hiding it for financial gain. So had people been questioning the norms, they would have seen it, before the company basically went belly up and took a lot of investors out. So, yeah, question the norms, because sometimes when the norms are hiding, people figure out ways to abuse those and hide things and should not be.

**Interviewer** 14:01

When I first saw your comment, I was thinking more about. for example, the acceptance criteria. would that apply to this statement? Like, there is an acceptance criterion, but we need to know, what is this for? What could be impacted by this change? What screens are we changing and why? Why are we doing that? so that our testing would be more efficient.

**P07** 14:31

Yeah, absolutely. And with that, you know, again, if you're not accepting, if you're questioning the norms, there may be things in those pages that seem or everybody says: Oh, those work just fine, but if you don't question the norm of those pages that are affected by what you're doing, there may be something introduced that would allow those pages to either be used in a way that they were not meant.

**Interviewer** 15:03

This is really good, because sometimes when there is a production issue, we go deep to see the root causes and see that was caused by lack of specification or the requirement or the acceptance criteria was not fully understandable. So even the developer or the test that they didn't understand, so for them was good, but since that information was not mentioned, and they didn't question it, we suffer the consequences.

**P07** 15:49

I do have, I do have a situation that happened at <company>, that the customer, which in this case, was <company> bank. That asked for a certain feature, and this was when I was relatively new at the company, but I was given the opportunity to test this feature, and unfortunately, as happens many times, people will name things a little too similar to be a good thing. So, these, I don't want to divulge too much. But these particular fields that they were looking to get a report on, there were two other fields that were named, very similar. So, as I looked at this disaster, they were like, Oh, it's these two fields. And as a tester, I looked at that and thought... And then it went back to my background of accounting, and I looked at those fields, and I thought, is that really what they're looking for? Because as an accountant, these two fields that they're asking for are not the ones that I would care about. And I know this is <company> bank they would want to care about these other two fields, not the ones that are being commented without asking for this particular feature. So, I did push back. And I asked specifically to those who were requesting that information. And I verified with them that these fields that were in the story were the correct fields that they wanted to record. And they said yes. And there was some other documentation that said it too. But I still had the question. And eventually I just said, You're sure? And they said, Oh, yes. Okay... And about six months later, when it was put into production, we got a call from those rather frustrated emails saying, this is not the information we asked. And, unfortunately for them, and for us, looking at it, and especially with my pushback on questioning what they wanted to, they had really asked for the wrong information, not even understanding it themselves. They had asked me wrong. And so that feature ended up being somewhat useless. Because they did not really understand what they were asking for, it was not what they wanted. Even with continued questioning, so yeah, pushing back in that case, could have helped. But it didn't because they weren't ready to answer.

**Interviewer** 19:20

This insight that you had was not really related to experience because you were new at the company, tight? So, it doesn't need to be someone who is like an expert to be able to have this insight.

**P07** 19:37

Yes, anybody could have done, If they were looking closely at what the norm was and question it.

**Interviewer** 19:56

The third one was understanding the product that you are testing but not be limited by its supposed constraints. Can you tell me a little bit about?

**P07** 20:08

Sure. That kind of goes back to the prior one when we've been talking about understanding the things that we are testing, and the products that it's going into, and what it is, and fits again with that experience that I shared, where I was relatively new, but I did understand the financial implications of those fields. And the two fields that they mistakenly asked for, and which one which one they want. But yeah, it is important to understand that, but then think outside the box of what they're asking for, to see what they're really wanting. That's, again, part of the scrum team needs to sit down together and determine and optimize what really is being asked, not always what they ask, but what their needs are and what would fulfill those needs.

**Interviewer** 21:25

So, when you said not to be limited by the constraints, it's in the sense of thinking outside of the box, I mean, thinking beyond?

**P07** 21:44

Yeah, so again, like in that example, being willing to look at it and say: Are there some other things that they might want to use, or that I should be testing? that should be part of what they're asking for in order to buy, or, in the case of, say, just a few... If you go look at a number field, and that's what's supposed to go in this field is a number, a phone number. And all that's supposed to go in that field is a number. If you just imagine the constraints are that number, you'll never test to see if you can put a letter in, or special characters. And it's important to do that, because somebody could cause some real problems in either the code or the database. There's always that lovely example... It's one of my favorites, that somebody told me years ago of a school using a database, and somebody who understood a little bit about databases gave their child's name as little Bobby drop tables, which seemed okay. But, when put into the database, that caused the database to basically empty and, you know? somebody would think of that and do it. So somebody is going to do something they will do so we got to think about it first, before it gets done.

**Interviewer**

Okay, the next one was be willing to say NO to allowing things to go not totally tested. Do you have an experience that you want to share?

**P07**

In my experience at McAfee, there was one QA in particular that a lot of people appreciated. But some did not tend to be a naysayer, he tended to say no, a lot. And rightly so. I learned a lot from him and his experience and his willingness to stand up and say no. Now, that didn't always mean that the NO was listened to. But he was willing to stand up against many, in some cases, managers that will like this just has to go. It's so important. It has to go in, we can't test all of it, so let's just have to put it in. The answer of no from this particular QA individual, and a few others, sometimes got listened to. And when they did, they were really happy that they listened. Because sometimes things are set up and put in faster than they ought to be, not for good reason. Just because somebody really wanted it. But it really could have waited. And sometimes they don't listen because there's so much pressure from out external sources. but they just go ahead and ignore the better judgments of some of the QA members, and they put the code in. And what you end up with, in almost every case, is some major bugs that cause problems that are worse than what was being put in place.

#### **Interviewer**

When you said, say NO when allowing things to not be completed tested. would that also apply to something for example... willing to say no when there is something that you think it's wrong, but someone like the developer that it's the normal behavior or even the businessperson say: it's fine... But you know that is wrong and have this ability to say no, I can't approve this...

#### **P07**

Yes, that is extremely important. Because your intuition comes from your recognition of how things should work. And when, when that comes to conflict with how you feel about a feature, you really need to take a second look. And, again, question the norms. And be willing to think outside the box and go and learn about what you're doing. So that you can come back and say, I did feel good about this. there's only so much that people will listen to you, when you say, I just don't feel good about. You have to then use your other skills and abilities to go find why you don't feel good about it. And be able to be able to and willing to stand up and say, now that is why. And sometimes again, they will listen and sometimes they won't. And typically when they don't, it ends up worse than what was put in.

#### **Interviewer**

It's really easy to think, to have this ability.

#### **P07**

Yeah, when you have leaders and sometimes, I know about others that don't... And it's happened in the past at <company> where the testers do not have to stand up to say No, and then just ignore and just do it. Yeah... It's important to stand up and say no, but sometimes just don't have standing in other's eyes, higher managers think it's more important to get over it. but it is important to be courageous and say no.



**Interviewer**

The last one seems almost like advice. It was like... in testing, If you can do it, you should do it.

**P07**

Like I said there, as opposed to in life, sometimes things that you can do, probably shouldn't. But when you're testing, if you don't test those things you can do but should not, then somebody else is going outside of, you know, in the customer area, or just a bad actor comes in and says: I wonder if.. like the experience of that school that had the database, where parents put in a little Bobby drop table.. they could do it, they probably shouldn't ... but a QA person should have said: I wonder!

**Interviewer**

Yeah, so it would be like, Oh, I'm testing something, and I'm like, oh, maybe I could test this scenario. But maybe it's not necessary... But if you ever think about that, you should test it. Right? Just to be sure.

**P07**

Yeah, it's very important.

**Interviewer**

Ok, so now think of someone that you have worked with before or you work now, and you don't need to say names. I would like to know why you think this person is great at her/his job? What are the things that this person does that you think it's really great and why?

**P07**

ok, there's a couple of people I can think of, and they all have very similar traits. They are willing and they work well... One of the things is they work hard, but also know when to stop. What is a nursery rhyme? I was All work and no play. I can personally make it so and so adult. And, I would have to say that that's very true. Also, the fact that they are adults and always think through everything that they need, because they're always working. They never let your mind rest to see other experiences to use for their work. There are plenty of experiences outside of work that are relevant inside work.

**Interviewer**

What do you mean? taking courses or talking to someone else in other projects?

**P07**

Yep, that would be right, in having taken the opportunity to relax and visit with other people in the company who may do the same job, or even people who do a different job, but that may be related to something that you're doing... take that time to stop

work and go play in a way that makes sense. Again, this is not necessarily about me, this is this is friends of mine, people that I know and look up to who are QA and this thing, and I've learned from. But I took some time and went to visit a friend on his birthday this Friday, and they've invited some other friends too. And I was reacquainted with a gentleman I haven't talked to for years and we had the opportunity to talk. And as you do with things that you care about you talk about what you care about, which in this case, is work and computers... We had opportunities and there were things I can remember from him in our conversation that I may well use in my work experiences.

**Interviewer**

I think it's kind of related to another trait that is managing our time well, you know? because sometimes we know it's important, we know a lot of things we can do, like people do want to talk, a lot of classes we want to take, but sometimes maybe we don't know how to prioritize our tasks to be able to do those things, you know?

**P07**

Yeah. Time management is definitely important, for a QA in particular because quite often there's not a lot of time, it's a lot of getting this done.

**Interviewer**

Okay. The last thing I just want to ask you to think of something that a great QA should not do or should not be... something that could be acceptable for some QA on average, but a great QA should not be this type of person or should not do these types of things. What comes to your mind?

**P07**

You know, thinking back on what we've discussed, I will have to say, not being courageous. You know, what we talked about was standing up and saying no, or being willing to test things that people are like: That's something you don't need to test, it always worked just fine. You should not test that... So, I think being courageous would be because you can be an okay QA tester if you're not courageous, because you can still test the things that are important, do your testing, but having that courage to stand up and be counted and say no, this is not right or Yes, it shouldn't be doing this... I think it would make it great QA because they can make effective changes for things that are important.

**Interviewer**

Do you have any other comments, any other things that you think it's important to say? feel free.

**P07**

I don't think of anyone else but thank you for inviting me.



**Interviewer**

Thanks so much for your time. I will stop recording now.