[02:43] PARTICIPANT 23:

Can you hear me?

[02:44] RESEARCHER:

Yeah, I can hear you now. Fantastic.

[02:46] PARTICIPANT 23:

I think there was something wrong with my... Give me a second.

[02:52] RESEARCHER:

Yes, sure. Take your time, sir.

[03:09] PARTICIPANT 23:

Yes, sir. I apologize. I think there is something wrong with my headphones.

[03:12] RESEARCHER:

No, it's OK. I understand. Let me start with some introduction. I will introduce myself and tell you what I do and why I'm doing these interviews and we'll kick off with the interview.

[03:27] PARTICIPANT 23:

Sure.

[03:28] RESEARCHER:

My name is [Deleted to preserve the participant anonymity] I am a postdoc researcher. I do my research in software quality. Basically, I try to understand how software teams manage to achieve quality and how do they implement quality in their processes. My current interest is Agile. And how software teams using Agile implement quality assurance and quality in general. [Deleted to preserve the participant anonymity] now I do have interests in Agile. Do you have any questions for me?

[04:19] PARTICIPANT 23:

Not at this time.

[04:21] RESEARCHER:

All right. How about if we start with some introduction? Can you introduce yourself and tell us what you do and talk a little bit about your experience?

[04:30] PARTICIPANT 23:

Yes, sir. My name is [Deleted to preserve the participant anonymity]. It's easier to pronounce. So, I have I will [inaudible] software testing experience. And I primarily worked in financial, like banking and like [inaudible] and healthcare domains. So, I'm pretty much an expert in those domains. And part of my responsibilities includes creating a test plan, and writing a test strategy, creating test cases and I also analyze, gather in business requirements and functional requirements from a business analyst like product owners and depending on what kind of methodologies. If I worked on Waterfall methodology, then I had daily interaction with my business analyst. Of course, in Agile methodology, you don't have a business analyst, so you have a product owner that you work with. If you have any product related questions. So as far as the testing tools that I used, I used HP Hierloom, [inaudible] lifecycle management. That's where you know, whenever I gather the requirements from the business analyst, and I upload those requirements in Hierloom. So, there are four different tabs in Hierloom, like requirements, test plan, test cases and defects.

[05:53] PARTICIPANT 23:

On the requirements tab, I pretty much link all that whenever I gather the requirements from the business analyst, I upload those requirements from Excel to Hierloom using a conversion method. After that I write test cases in a test plan folder. All the functionalities what they want to test, whether it's a log in feature or a name and a password. And we also have a test cases folder for the test run. So basically, you execute them in this cycle, like cycle one, you execute these list of test cases based on the functionality that was delivered by your team developers and your tech leads. Based on like an all your inputs and findings, you pass them, or you fail them in the folder. And we also have a defect tracking folder. So basically, you log all those defects and you assign a criteria, whether it's a critical, high, medium or a low. So, it is as part of the testing team, it is my responsibility to identify how we know how I defined a particular defect as a critical defect or a high defect, high priority, or medium priority.

[07:12] PARTICIPANT 23:

I assign those defects to the appropriate developer and wait for the resolution. And when it comes back, for instance, if the developer says, this issue has been fixed, please go ahead and retest. And that's where the regression testing comes into picture. So, I read on the test case to make sure the functionality is working as expected so that I could close the defect. However, sometimes you may have to run the test case several times because sometimes when the developer's fixed fix this feature, the existing functionality could be broken. So, you have to run other test cases to make sure there is no impact as a result of this fix. So those are my day-to-day responsibilities I was involved in, Like in my professional summary, creating test plans and executing the test cases and the business analyst you know, when I subject my testing strategy, or test plan to the business team so they will give a testing sign-off saying that you are good to go and we ready to implement this product.

[07:47] RESEARCHER:

Ok, great. Let's define quality first? How do you define quality in the context of agile software development?

[07:57] PARTICIPANT 23:

You will get different answer from different people that for sure. For example, we QAs focus on finding bugs. For us software quality is free of bugs and meets the requirements. For developers, it's code quality and design. But as a team, we collectively thrive to improve our process to deliver quality software.

[08:17] RESEARCHER:

Ok, great. Let's talk a little bit about Agile. What is your opinion of Agile?

[08:24] PARTICIPANT 23:

Agile methodology is, let's say, very, very critical role as far as the software development lifecycle, because it involves short iterations for instance, I was involved in PI planning in Scala program increment. So, in programming it typically consist of five sprints. Each sprint lasts for typically two weeks. So basically, you work with your team is a self-governing team. So, you have your product owner, Scrum master, your tech lead, developers, and testers. So typically, no more than like seven to eight people in a team or sprinting. So, in each team, so basically you know what direction you are heading to, because we have a user story that is defined by your product owner. So, for instance, as part of my testing job, I accept that user story. And I started working on that like let's say I give them, prior to the sprints start, I give them the...okay, it takes me like fifteen hour testing effort for this user story to be completed. So, I provided those number of hours per sprint. Again, from a developer point of view, he would give you an estimate. He says that he might say that he takes like forty hours' worth of development time and from a technical point of view, he says I need like six hours each sprint to review all the specifications in order to go through the ad hoc testing and all those. So basically, with Agile, you work in a team, a self-governing team.

[09:54] PARTICIPANT 23:

You responsible for creating the stories and providing the estimation hours including like how long it will take the hours and any dependencies. If you have any dependencies with the external team or internal team, if there is an impediment, you should raise all of those things that's part of their sprint planning. So ideally, there is a huge benefit being in Agile because you should be able to deliver a product in short iterations and in every sprint, you have something to deliver, it could have a big feature or a small feature or enhancement. You always have something to deliver to show in an Agile environment. And you also get daily feedback from the product owner because as part of the sprint team's responsibility, we should demo our product to the product owner to get their constant feedback. Is this the product that you are trying to launch, please provide the feedback before we could deliver. So, we obviously get a daily updates and daily comments and daily reviews by the product owner. You know how the product behaves by adding more features and all this. So compared to the Waterfall methodology, because you always have an opportunity to get the constant feedback from a product owner. And so that way you could work on that by updating the developers.

[11:10] PARTICIPANT 23:

I'll give you an example. The last project I worked for, [Deleted to preserve the participant anonymity], it's banking, we have a function like let's say as a user, as a consumer logging into the product with my username and password. We developed a feature called One Time Pin. Basically, that's an extra security measure. So, make sure the person that is logging into the system belongs to the actual consumer. So, when I log in with my username and password, I should get the one time pin page. It will give me multiple options like would you

like to send your one time pin to your e-mail address, or to your phone. It gives them multiple options. So, this is part of my user story like as part of our team, we should deliver that functionality at the end of the sprint. But like in the beginning of the sprint planning, the product owner will give you a high level overview. They would just say developed the One Time Pin with the e-mail and the phone number. But he doesn't have any detailed specifications, like in the Waterfall methodology where everything is a detailed and well documented. In Agile, everything is like in a high level. And they just give you a high level overview like how the product works. So, we develop the feature and worked on the feature. And then we have a sprint review and we've demoed our product to the product owner, and he said, yes, the page looks good enough. This is what I expected, but as part of the security measures, we don't like to display the end user email address or the entire phone number.

[12:46] PARTICIPANT 23:

However, this was not defined at the beginning of the sprint. Like I said, in Agile they'll just give you a product overview, they don't go into the deep technical specifications. In Agile like midway through the sprint, our development team has to go back and reupdate their record so to make sure the email address and phone number are masked, instead of displaying the whole thing. Again, as a testing team, my priority is to go back and prioritize the test case and add more test cases based on the product owner's update and intent. Like I said, in the end we should get the product as expected so that's how Agile works and you should always get constant feedback. You should always be in a position to update test cases or update the code, so there are several factors that could change.

[13:39] RESEARCHER:

Fantastic, thank you. I'd like to follow up on a few things. The first thing is you mentioned you accept the user story. So, you are in engaged from the beginning of the process, as a tester?

[13:54] PARTICIPANT 23:

Yes, so basically prior to the sprint planning, we have something called program increment. So, PI planning is done every three months, so every three months it consists of different teams, all the integration teams that's working towards delivering the product. You have like team A, team B, team C and they should all be part of the PI planning. So, in the PI planning they have a product manager. He'll give you a product intent. Like, this is the product we are trying to launch. In my experience, as I told you earlier, the product is all about delivering a One Time Pin to the customer feature. So, team A is responsible to develop that web page within that sprint and team B's responsibility, let's say whenever he sends a request, send pin request, so you get the pin back to you, the customer. So, team B is working on the send pin request. Or team C is working on the access pin request. Team B is working on the phone number request and response. Basically, all the teams will be part of the PI planning.

[14:59] PARTICIPANT 23:

And each team has an assigned user story. So, okay, go ahead and work on this feature like team A develops the web page to the customers. Team B go ahead and work on the send pin and accepting. Team C request pin so different features were accepted by the different teams. And all these stories were given by the product owners. Basically, they assign all these products to the different teams based on the needs.

[15:27] RESEARCHER:

So, how does it make you feel when you are engaged from the beginning? This is unusual to the traditional method where the QA is at the very end, sometimes it's forgotten. How does it make you feel in this environment, you are engaged from the beginning?

[15:47] PARTICIPANT 23:

It makes me feel very confident about the product. I'll tell you why. Because it starts from the PI planning, all the way to creating and accepting user stories. Because I know what direction I'm heading to. I know what's my user story, I know what my focus is. I know what the deadline is. So, I have to work everything in, and I feel actually happy because I have constant interactions with my tech lead and product owner. So, any time I have any questions regarding the product, or any technical specifications, I could directly engage my technical lead unlike in Waterfall methodology, you start in a very, like you don't interact with your technical lead, to be honest. You just start working on your requirements and start developing the test plans and test cases. But in Agile, that's the benefit because you get to know more about the product, you get to demo the product to the product owner, the rest of the team and also a wider audience. You know you have a tight timeline to deliver the product, you always have something to deliver at the end of the sprint. So that is what being a QA, I like about it.

[16:58] RESEARCHER:

Fantastic. How this help you achieve quality?

[17:02] PARTICIPANT 23:

The more I know about the business needs, the better I test and the better I test then less bugs. I forgot to tell you one more thing. So, we have two things. Our sprint consists two weeks. Midway through the sprint we have a sprint review to see how things are moving, are there any impediments, things like that. And we also have a sprint retrospective at the end of the sprint. In the retrospective, the Scrum master will ask us everything like what things can we do to make things better, is there anything we can do better, what are the impediments, what are the challenges that we faced during this sprint. Do we have any dependencies from an external team, we didn't get responses on time, things like that. There is always a sprint retrospective and apart from that you also have daily stand-ups. Sprint stand-ups. Every day the team should gather in a room and do updates to the team saying, okay, there's a user story I worked on yesterday, I spent six hours and out of six hours, I worked four hours on the test execution and two hours working in retesting the defects, things like that. And the remaining hour I worked with the external team to get the test data setup. And there were no impediments, like depending on your daily tasks, sometimes you have impediments, sometimes you don't. But it is your responsibility in the team to raise an impediment to the Scrum master. As a Scrum master, it's his job to fix and facilitate with higher management and other teams to get all the required things that we need.

[18:34] RESEARCHER:

You mentioned that you are a self-governing team. Can you elaborate a little bit on that?

[18:42] PARTICIPANT 23:

Yes, the team consists of product owner, Scrum master, technical lead, developers, and testers. So unlike in Waterfall methodology you had a project manager, he or she will oversee the project. They will give you the hours and they will tell you, okay, you need to complete like thirty hours of development or something. But in self-governing team, in Agile, prior to the sprint starts, we know how much time it will take as a team effort, as a collaborative effort. We know how long it will take to develop the product, to test the product, to deploy, to maintain the product. So, we have nobody to oversee. It is our responsibility as a self-governing team to work on the product so like I said, at the end of the day, the product owner, and Scrum master they will be in touch with the rest of the team and planning. So, like this is the progress that we are currently making on this functionality and so basically you don't report and you're not responsible to anybody except your team as a self-governing team on the daily tasks that you work on.

[19:55] RESEARCHER:

You mentioned some few values of Agile. I'll list some of them and I'll ask you a question regarding that. You listed frequent deliverables, ongoing feedback, collaborative effort, and accountability of the team. How do these values help the team to be more productive and to deliver better quality?

[20:27] PARTICIPANT 23:

It should be like to develop the better quality, like being in an Agile environment, because you get constant feedback about the product does not mean you're going in the wrong direction. You may be going on the right path however sometimes the product owner like prior to the sprint, he may not be clear to you how the product works and how the feature works. In the example I gave to your earlier, the One Time Pin when you login with your username and password, all the requirements we got from the product owner was just develop this web page just with your phone number, you give an option to the user as either phone number or their e-mail address. That's part of the requirement and we developed it and tested it, and everything works as expected. As part of the sprint review, review meeting, we went through the sprint session, the product owner came back and said, after we presented the demo, yes this page looks good but as part of the security measures, we don't like to display the whole characters or whole phone number to the user, so we should mask it with asterisks.

[21:36] PARTICIPANT 23:

But this wasn't clearly defined as part of the product and even he didn't know because when he was given the requirements by the product manager, that's all he got. But when we provided the demo, that's how we got the feedback from the product owner. Like we should have more features that are part of the security measures. So, it wasn't clearly defined. In Agile you don't have the product clearly defined. You start working on it as soon as you get it, the user story. This help us to meet the PO needs with less defects because of the ongoing feedback.

[22:03] RESEARCHER:

How do you work with that lack of detail. The user stories are very high level, how do you work with that lack of detail in the process?

[22:15] PARTICIPANT 23:

So that's where the technical lead comes into the picture. Whenever the product owner gives a feature like the login functionality, for example. All we know is develop the web page with the login feature with a username and password. But from a testing point of view, what I would do is write a scenario. Scenario one, enter the username and enter the password and click on Login. I expected it to be that the user lands on the Account Summary page and all the products should be displayed that the user is eligible for. However, there is a lot more dev to that. The technical lead will think of all the specifications. The login does not mean you just login to the application, it could be like several different aspects like what happens if you don't click or you leave the username and password fields blanks and you click on the Login button, what will happen.

[23:09] PARTICIPANT 23:

We know what will happen. Because we get an error message stating you need to enter the username and enter the password in order to login. However, in the product it's not even different like what kind of error message will we get. So now we have a tight deadline, we should start achieving so the developer will start displaying the error message saying that, please enter the username and password. The reason is rather than not displaying any error message they start developing a prototype. And when we like have to demo the product to the product owner so we present the error message to the product owner every day and the product owner then will come back and say, yes, this error message looks good but please add in one more statement. So, the developers just have to go back to the code, and they add the extra statement. And that's part of my testing responsibility, all I had to do was go back to the test cases that I wrote and what kind of error message is supposed to be displayed, and I just add the extra error message.

[24:15] PARTICIPANT 23:

So, that's when the regression testing comes into the picture. We use automation testing tools like simply run everything and make sure the existing functionality is not impacted as a result of this code change.

[24:26] RESEARCHER:

This highly collaborative environment, does it help achieve quality?

[24:33] PARTICIPANT 23:

Yes, definitely because as part of the PI planning, program increment plan, you know what you're focusing on, your project focus. If you have any impediments in the beginning of the PI, you should let them know like I need a resource, like I need one more developer from the team. Because in order to deliver this functionality, we at need at least three developers and right now we only have two developers. In order to highly elaborate all these resources, we need to add one more developer. You could help us to borrow from another team or help us find a resource to help the project, so that will help us to highly focus on the product that we supposed to deliver.

[25:20] RESEARCHER:

Great, thank you. Next question, what do you do to assure software quality in Agile processes?

[25:31] PARTICIPANT 23:

Yes. So, make sure we have exit criteria before we close the user story. Because at the beginning of the sprint, we accept the user story and start working on it and provide the number of hours. At the end of the user story, we have some guidelines. In order to close this user story, what are the guidelines, what are the criteria needed to close the user story. My criteria as a quality assurance in the Agile environment, make sure the user story meets all the criteria including the business team's criteria and the product owner's criteria. All the use cases have been met, all the test cases have been executed, there are no critical, high, or medium defects. It is fine to have a low priority defect for instance the look and feel of the application.

[26:15] PARTICIPANT 23:

We could always defer the defect to the next sprint because it's not a showstopper. It shouldn't stop customers from accessing the application. So this is the criteria, the first criteria I look for is to make sure the user story accepts all the guidelines, all the test cases have been passed, we get a testing sign-off from the product owner, all the test demos have been signed off by the product owner. There are no critical, high, or medium or showstopper defects. That's how we achieve that quality in the product process in Agile.

[26:49] RESEARCHER:

Do you think this process produces quality software?

[26:54] PARTICIPANT 23:

Most likely yes. But in like Waterfall methodology you have longer iterations, it could take six months to a year to deliver a project or it could be four months. However, in Agile methodology, it’s an ongoing demand from the customer. They would like to see each feature on the web page, they like it to be more user-friendly features, so it gives us a quicker time to deliver part of the product to the customers. To me the customer needs are business needs. That's the difference between Agile and Waterfall.

[27:30] RESEARCHER:

So, this fast delivery, does it create pressure on the team?

[27:36] PARTICIPANT 23:

It does create pressure to some extent. But it also gives you the opportunity to deliver something quickly. That's why we have the sprint retrospective. Each PI planning, we have five sprints, all the way from sprint one to sprint five. In sprint one, when we deliver something at the end of the sprint, we have a retrospective. So, we knew what went wrong with the product, we knew what can be done to get it better. That's what the retrospective is about. We learn from our learnings in sprint one and we can implement that into sprint two.

[28:09] PARTICIPANT 23:

For instance, in sprint one, we have a dependency on some other team to get the test data in a timely manner. But some days we didn't get that test data due to the environments being down. So, that's an impediment, right. That will be discussed as part of the retrospective and sprint review. What we do before we even start the sprint two, what we do, we'll start getting everything ready in advance. So, make sure like prior to the sprint starts so that way we should be able to achieve and deliver without impediments in a timely manner. As soon as we finish the sprint retro, so we will start working with the other team to get the bandwidth and to make sure we have the user story so we can setup the test environment well in advance before we even start the sprint.

[29:02] RESEARCHER:

There are two things I'd like to follow up. The first thing is you mentioned there is ongoing learning about the product. Does it help you, this learning of the product, does it help you in assuring quality and doing your job?

[29:25] PARTICIPANT 23:

Definitely because the feature or the product that we are delivering is in iterations. The cycle. Each time when we deliver something, of course we get the feedback and we know when a test case went wrong from a development point of view or from a quality assurance point of view. This features looks good, but this has not been tested, that's one of the feedback we might get from customers. But in the beginning of our Program increment, this is out of scope, this is not even discussed that this feature needs not be tested because they said it can be pushed into future releases. But that's not how each time we do it because in the end when you deliver something to the customers, you'll always get reviews and updates saying it would have been better if you have delivered this product along with this feature.

[30:15] PARTICIPANT 23:

Because right now, this feature is not working. But like I said, being a team, we did not commit that specific user story to deliver in this sprint. So, we can take this lesson as a learning lesson when moving to the next sprint, so after we delivered sprint one, we can move to sprint two and deliver this and we can talk about, this is the feature that we did not deliver but this is intentional because we knew we didn't have to deliver this specific feature because this is not part of our user story. But what we can do is, we can add miscellaneous tasks from the next sprint onwards. So, although it's not part of the deliverables if you have any bandwidth, whether it is from a development team or the QA team, we can say, okay it takes me twenty hours of testing effort in this sprint, to deliver this product but I could also allocate like an extra four hours of miscellaneous hours to work on extra features although it’s not part of the development or testing deliverables.

[31:23] PARTICIPANT 23:

So, we can consider that, so in that way we could deliver a specific product with the features even if it’s not part of our PI planning. So that gives us additional time to work on some more products and get to know more about the product for upcoming sprints. So, you can deliver the product well in advance prior to the deadline.

[31:47] RESEARCHER:

Okay, fantastic, great. I would like to follow up on another thing. You talked about Agile ceremonies, you talked about retrospective and sprint reviews. How do these ceremonies help in achieving quality?

[32:07] PARTICIPANT 23:

Can you please repeat the question again?

[32:10] RESEARCHER:

You talked about Agile ceremonies, you talked about retrospectives and sprint reviews, right?

[32:20] PARTICIPANT 23:

Yes, sir.

[32:21] RESEARCHER:

How do these ceremonies help in achieving quality in the process?

[32:28] PARTICIPANT 23:

Definitely. It makes a huge impact on overall quality because like I said, for instance, you deliver something and during the ceremony, you display your product to the product owner. However, to make things even better, to make it much higher quality, you could develop an automation suite rather than doing your manual tests, doing the regression testing over and over. It might take like twenty hours of time to do the manual testing. At a result, the quality might be the same but there is always a chance you might miss a scenario or two so that's part of the regression manual testing. But the benefit of automation testing when you develop an automation tool, like if there's a code change, if there is a functionality change, you just go back to the script and just update those scripts only. All it takes it just instead of running twenty hours of manual testing effort, automation it takes like a couple of hours. You should be able to achieve the quality much quicker. And you can identify the defects in a timely manner. And you should be able to report to the development team and that gives them enough time in the sprint to fix the issues and get back to you. So that way, you can still re-execute the automation suite and then it’s good to go. So, that's one way to achieve higher quality. Develop an automation test suite so that way it will save you some time and also time to find missing critical functionalities.

[34:06] RESEARCHER:

Great. Can you share with me a positive story about Agile and its ability to produce software quality?

[34:18] PARTICIPANT 23:

You mean a live example?

[34:20] RESEARCHER:

Yes.

[34:23] PARTICIPANT 23:

So, you mean like how we achieve quality, right?

[34:27] RESEARCHER:

Yes, an example. Share with me an example.

[34:32] PARTICIPANT 23:

Can you elaborate a little bit on this question, please?

[34:35] RESEARCHER:

Yeah, of course. Can you share with me an example where Agile helped in achieving quality?

[34:45] PARTICIPANT 23:

Okay, I got it. Agile methodologies is a very important, active role in the SDLC, the software development lifecycle. You should be able to, you know, rather than wait to the end to know your product intent, your specifications. You get them well in advance, as part of Agile, so basically you know what direction you're heading in to achieve the timelines. Because as soon as a user story is given to me or given to anybody, like as part of a self-governing team, we know the deliverables. Like this user story has to be done within a couple of hours because we know we have so many hours in the beginning of the sprint. So, based on the timeline, you know you have two hours of time to work on this user story and to complete these tasks.

[35:39] PARTICIPANT 23:

So, you should be able to identify all the use cases pertaining to the story. For instance, being a quality assurance team, what I would do is once I get the user story, to improve the quality measures, once I get a user story, I define all the acceptance criteria. This is the acceptance criteria in order to successfully continue or complete the user story. I need to make sure I have a test environment ready and make sure I have different browsers, like Chrome, or Firefox or Microsoft Edge, and mobile testing devices like iOS, Android, so things like that. So, I make sure achieve I can achieve the overall quality like with all this acceptance criteria. In order to continue with my user story, I need to make sure this acceptance criteria is met.

[36:31] PARTICIPANT 23:

My testing environment should not only focus on a web browser, I should also focus on mobile because that's part of our next generation technologies. More and more users are utilizing the mobile technologies. Though it’s not part of my scope, in order to achieve the overall quality, I should also engage to test in different mobile browser labs, like source labs, you can gather any type of devices that you would like to test. So, that will give me overall quality achievement. Like I'm able to test all these features. Like in the beginning of the

sprint, the product owner said go ahead a test on Chrome browsers, web browsers, you know, whatever devices that you want.

[37:21] PARTICIPANT 23:

So, for instance, like on an iWatch device, but I don't have any other devices. So how do I achieve this quality. So, the best way is I give you an example, like sourcelabs.com where you can go there and configure any mobile device, like [inaudible] like different types, like Android and all those. You should be able to test in different versions like version six, version seven, version eight. So, that way you should be able to achieve the quality by testing that way. Because some customers have version seven, some customers are still using version five, some customers are using version eight. So, make sure as part of your QA, as part of the Agile process, you are able to go back to the previous versions too to make sure you can test those versions to achieve the highest quality. You need to make sure in the end, since you are delivering the project within every two weeks, no customers should be impacted whether that customer has been using that mobile device for a long time or whether he has been using a specific Chrome browser for the last three years, he never got the chance to upgrade. But it is my job as part of the QA, in order to achieve the highest overall quality, I need to make sure I had to go back as far to the oldest versions that are in place and to the current versions.

[38:38] RESEARCHER:

You seem like you are a very mature team in doing software. You do have strong practices in place. Do you think that this maturity has helped in getting Agile right?

[39:00] PARTICIPANT 23:

Definitely. Like I said that retrospective definitely. That ceremony will help me. All the team members need the feedback, it's not just from my overall team like okay, we delivered this feature. You shouldn't work like that in Agile. You work as a team, you each play a key role. Every team member within the sprint has their own responsibility and you should share your opinions, what your thoughts are, what can you do to makes things better, that's how the ceremony comes into the picture. When you interact with your Scrum master and product owner, how to facilitate this sprint moving forward better. Definitely, like being detail orientated, you should help the team, not just you, to deliver the products in a timely manner.

[39:56] PARTICIPANT 23:

And also, you don't have to blindly accept a user story. Whenever you get a user story from your product owner, you don't have to accept it because you have to go through, okay, what are the conditions in order to accept this user story, are there dependencies from another team. What happens if the team is already occupied with their user stories, they don't want to allocate their bandwidth to us, so we have to consider all those factors, not just like, how much time do you have left, how many dependencies do you have from the other teams like for the test data, or setting up the test environment, things like that. So, you should always look at the overall project point of view, not just your team. The project consists of several teams. You should always consider all of these facts depending on the bandwidth of the team, other teams, how much can you interact, what happens if the environment is down, how quickly you are able to resolve. What happens if the issue is not able to resolve, can your product owner or Scrum master escalate this issue to get it done in a timely manner. In the end, you don't want this impact on the deliveries.

[41:01] RESEARCHER:

You seem like you have as a tester or a QA, you have a voice in the process. And you seem to be happy because of that.

[41:14] PARTICIPANT 23:

Definitely, I feel a lot happier because being a QA, it may not be funny to the developers, but my job is to find as many defects as I can. That's part of the product intent, they only tell us, okay test these features. But I would like to test additional features which is not even in scope, so that way I give my feedback well ahead of time to the developers. Okay, this functionality we not supposed to test but I would like to go ahead and test even though I know it's going to fail because it's not part of the testing scope. It's out of scope. But I would like to test this feature, and these are the possible use cases that I could develop before the next sprint. So, that way, this might be useful to you. These are the possible ways I'm going to break the code so I can find many more defects. In order to save time for you, you've already started developing the code when the sprint starts.

[42:08] PARTICIPANT 23:

Just focus on those primary features, and these are the use cases I'm going to provide to them so that gives them enough time and also saves them time while they are coding the functionality. So, they can simply go through the use cases that I've written from a QA team and then make they cover all those features. So that gives them enough time to deliver the code to the testing team for us to test so they don't have to keep back and forth, saying okay this functionality has been delivered, please go ahead and test. Like I said, if they go through all those use cases and run them like a sanity test, it's not like a detailed kind of black box testing. The developers can only test to come extent. So, they can use the high level use cases scenarios to make sure its covered and they can deliver. The chances of finding defects, like critical defects are very few.

[43:06] RESEARCHER:

There is something very important in software development, which is the relationship between the QA and the developer. Can you talk to me a little bit about that and how Agile makes it better?

[43:24] PARTICIPANT 23:

Most of the projects that I worked in the past, a lot of times there is miscommunication between developers and tester. Starting from the user story all the way to when we raise the defect from the QA team. So, the developers come back and say, this is not a defect, this is what is expected. They might be right from a development point of view because they have their own specifications but as part of the QA team, we do have our set of guidelines because after development, the quality comes into the picture before we deliver to the customer. So, we need to make sure, we need to check everything, all the use cases whether its positive cases or negative cases, security testing, testing in different browsers, and all those features. And we should be able to create all those use cases in a format they would like, the developer.

[44:16] PARTICIPANT 23:

Okay these are the features we have tested as part of the QA team. These are all the different criteria that we use. These are the different test cases that we used. And I'm just testing, not from a QA point of view, I'm testing from an end-user point of view, I'm thinking like the end customer to this banking. And I have every right to test this feature, any use cases that I can think of, and I can report it back to the developer. However, there should be an understanding between the developer and QA on an ongoing basis because when I develop the user story in the beginning of the sprint, the testing team will give their timelines. It will take me three hours to develop the test cases and what you do, you write all those scenarios and at the end of the day, what you can do is have a meeting with your developer and tell him these are the possible scenarios that we could test. In order to save time for our team, you could also focus on these use cases in addition to what we already have. Because the developers have their own use cases as part of their unit testing. But as a QA team, we should provide a detailed plan, like these are the positive scenarios and these are the negative scenarios that we would like to test, and we have found some findings in these test cases.

[45:34] PARTICIPANT 23:

We should always have a daily conversation with the developer, like when they are developing, it doesn't have to happen every few hours but at the end of the day, you should a meeting to saying, okay these are the use cases that we are testing or that we tested today. And these are the defects. And reporting defects, sometimes you don't have to report some defects in Jira or a quality center like an ALM, what you can do, you can resolve them when you have a developer sitting right next to you. Then you can explain to them, okay this is the defect that I found. He can go ahead and fix it, so that way, we don't have to enter that defect into Jira or ALM, just report to the developer and they work on the task and then getting back to you with their comments. So sometimes on the spot you are able to work with the developer and resolve that issue. That's working in Agile.

[46:25] RESEARCHER:

Yeah, great. You've been talking very positively about Agile. However, it doesn't go well all the time. I'm sure in your eleven years' experience you may have a bad example or a negative story that you can share with us?

[46:45] PARTICIPANT 23:

Yes, I will also repeat this thing that I talked about earlier. In one of the products that we launched, a couple of years ago. The product intent was very clear, and they had very tight timelines for every sprint to deliver the project. However, there were some features that were out of scope, details and dependencies or various factors and things like that. However, the product owner, or project manager they came back and said, okay, we don't mind delivering this product to the customer without adding these extra features, so we had to follow their guidelines as part of the team. We worked as an Agile team, we developed the product, we tested it and we delivered. All those functionalities were working fine as expected and in a timely manner. In the end, we got a lot of feedback

from the call centers. So, when the customers ran into issues, we don't interact with the customers, we interact with the call center people.

[47:47] PARTICIPANT 23:

So, the call center got a lot of feedback from the customers getting a lot of negative feedback. It's not because of the quality of the web page, it's because of how the features are developed and put in place for the customers. But a few customers came back and said, this page is not user-friendly. Though we are able to login and we are able to access the icons, this is not user-friendly to some people who are disabled, and they are trying to get accessibility features. We got a lot of negative feedback from the customers. That was one thing that we discussed in the PI planning, when we had the PI planning, we said that yes, being in an Agile environment, it is very important to deliver the products in a timely manner, very quickly, like every two weeks, every month to the customer. The quality is being important, but we also need to consider the quality but delivering the products quickly we're missing key elements, quality of the product not just the quantity, like hey, are you able to do it on a timely manner. Yes, we did. What about the quality, like how much does it impact the end user, how much negative impact does it have to the end users. We should consider other aspects.

[49:05] PARTICIPANT 23:

So, moving forward what we can do right, in order to reduce all these negative effects, let's break them down into features like let's say, if you're delivering a product with ten different features, you develop five features at a time. So that way we can get around this feedback so instead of delivering every three weeks, we deliver every two weeks so that we have five features and then focus on delivering the next five functionalities of five features to the customer so that way, the business won't be impacted and the customer should be happy. Otherwise if you simply just delivering the whole project without looking into the quality, right, that will definitely have an adverse impact on the business. My priority is to focus on the deliverables as well as what you can do and what you cannot do. If you cannot do a specific feature, you don't just deliver it, just go ahead, and deliver what you can that's accepted in the user story. That way you can avoid all the negative feedback, or most negative feedback from the customer.

[50:10] RESEARCHER:

So, as a conclusion, as a team from this example, as a team, you learn your lessons and you took some actions to improve the process and your practice and you put those actions in place, right?

[50:27] PARTICIPANT 23:

Yes, sir.

[50:29] RESEARCHER:

Thank you very much. I've run out of questions, do you have any questions for me?

[50:35] PARTICIPANT 23:
Yes, it would be great, I don't like to ask you this question.

[50:43] RESEARCHER:
Thanks for your time. Bye.