[00:45] RESEARCHER:
Good morning, how are you today?

[00:47] PARTICIPANT 29:
I'm doing great. How are you, Researcher?

[00:49] RESEARCHER:
I'm very well. I’m just going to start by introducing myself. After I introduce myself, we can start with the questions. How does that sound?

[01:00] PARTICIPANT 29:
Sure.

[01:01] RESEARCHER:
My name is [Deleted to preserve the participant anonymity]. I do my research in software quality. I try to understand how software teams produce software quality. I do have other research interests like open source software and Agile in general. So, the purpose of this interview is to gather people's perspective. We do believe that peoples knowledge bridges the gap between theory and knowledge so that's why I'm gathering data using interviews. So, I interview people to get their perspective and their knowledge regarding certain problems or perspective. So, that's what I do. Do you have any questions before we start?

[02:04] PARTICIPANT 29:
No.

[02:04] RESEARCHER:
Alright fantastic. The purpose of the first question is to get to know you. Can you please introduce yourself and talk about your experience briefly?

[02:19] PARTICIPANT 29:
Sure. I am [Deleted to preserve the participant anonymity. I have done a bachelor in communication and I passed out in 2006 and then I started working as a junior developer in one of the dev teams where we are responsible to developing web and mobile apps. I joined several companies and gained my experience. After about 12 years plus experience in total and Scrum in mobile apps, developing web apps, testing all that stuff. My experience on processing, I worked as a Scrum master as well where I managed multiple teams and delivered end-to-end products.

[03:12] PARTICIPANT 29:
Then I was responsible to kind of get the requirements and to make sure the requirements and everyone in the team is on the same page as part of it. And then I was responsible to set up the planning meeting and once all the team is agreed upon, I'll supervise the story to get a sprint and then run the entire Scrum. I did handle the release committing meetings and all. So, I took care of all the agile ceremonies as well.

[03:56] RESEARCHER:
OK, fantastic. I will go back to some questions in your experience.

[04:01] PARTICIPANT 29:
Sure.

[04:14] PARTICIPANT 29:
I'm sorry for the background noise.

[04:16] RESEARCHER:
No, it's okay, we can live with that. OK, how do you define quality in the context of agile software development?

[04:20] PARTICIPANT 29:
I’ll start with no defects. But a messy code can have no defects. We focus on code quality or clean code. We assure it by code review and having guidelines and standards. The design is another aspect of quality. It has to be sustainable.

[04:46] RESEARCHER:
So, in your resume, it says leading Scrum teams for QA efforts. How does a QA team work in a Scrum team?

[04:51] PARTICIPANT 29:
So, Researcher there are different rules in Scrum, right?

[04:56] RESEARCHER:
Yeah.

[04:57] PARTICIPANT 29:
One master, product owner, then developers and project managers and the QA. So, QA is responsible to test all the acceptance criteria into the user story has been properly launched as

part of story deliverables and being the QA, we used to test areas outside of the user story. So, the product story we are delivering that there could be impacted area out of the story. So, we do test regression and make sure everything that is already existing into the software is up and running and they are not breaking our code.

[05:17] RESEARCHER:
So, how does it work? Are the QA engaged from the beginning? What's their role? Can you take me through the journey of a user story and how the QA are engaged?

[05:34] PARTICIPANT 29:
Yes. QAs is definitely engaged from the beginning because when the grooming meeting are being set up, QA is part of it. So, QA should have the understanding of the user story like every other team member. So, everyone should have to be on the same page as part of user story understanding. So, when it comes to testing, I won't waste a lot of time to understand the user story. When the user story pulled into the current sprint backlog, the current stream, the developer who starts coding and QA will start creating the test case. Once the story is out, the QA need to execute his test cases and make sure the user story acceptance criteria are being coded. Both the developers and QAs are in sync, they understand what to deliver and how to test it.

[06:27] RESEARCHER:
So, does this mean less bugs?

[06:30] PARTICIPANT 29:
From experience, yes. When the QAs and the developers collaborate closely, we see less bugs.

[06:30] RESEARCHER:
So, at what stage of the process you start coding? So, given a Sprint. Let's assume there is a Sprint one and Sprint two. In Sprint one, at what stage do you start testing?

[06:46] PARTICIPANT 29:
OK. So, let me give you a little brief answer to this question. I'll say for example like you mentioned, there is maybe I'm releasing my product after three sprints. Every sprint is maybe three weeks of sprint. Maybe after three months I'm releasing my end project. Now maybe every single Sprint, which is three weeks, I am pulling maybe ten user stories. Now once this story, like everyone is on the same page I used to, the Scrum master used to set up one criteria called the definition of ready. So once this definition of ready is set, the developer would start coding the story. QA will start creating a test case. There could be multiple developers into the single

Scrum team. So, stories can be divided into various developers now once developer is done with that coding, they can mark a story read to test. QA will pick the stories and if there are test cases for the users to execute the regression. Then call out that if that story is good to do duly or not.

[08:17] PARTICIPANT 29:
Now there is some criteria we had set up for our definition of done. Which is what we call a DoD in short. I'm just going to explain couple of criteria of what we set up to call out a story as DoD. First all the acceptance criteria has been implemented. Second, there is no P1 and P2 defect. There are minor defects which are still good to call out a story as a DoD. Stories can be delivered into the development environment. So, we used to kind of demo our work. So, what we've done is part of sprints.

[09:14] RESEARCHER:
This sounds a little bit of waterfall don't you agree with me?

[09:20] PARTICIPANT 29:
I wouldn't. I'll tell you why. Agile is incremental, right?

[09:26] RESEARCHER:
Yes.

[09:27] PARTICIPANT 29:
So, in Agile, how we did work is we used to code step-by-step. So, in the sprint, if I'm able to complete one feature or couple of features, those features are good to go and then I can start working on the next set of features. Like I said Researcher, it would depend on companies on what sequencing they would like to set. So, what most of the companies are doing is to release frequently to the production. Alright, so once a single sprint is done, they are good to be released according to the production. So again, here we are doing it through incremental way. So, I don't think that could be a waterfall. I would rather speak about Agile and the only reason being, we have taken one feature to deliver that is done. Okay, put it aside, pick up next feature. Complete that, once integrated, test all the features together.

[10:34] RESEARCHER:
OK. So, let's go back to my scenario. We are still in Sprint one and you start testing. What does the developer do when you test because practically, they finished developing and you started testing so it's step after step? That's why I call it waterfall. So, what do they do when you are testing? How do they invest their capacities?

[11:11] PARTICIPANT 29:
Yeah, so there's a really good question, Researcher. So, there are two scenarios of it. We have to face the situation initially. What we've done, we had to work out stories. What is work out story, there are some stories which are already in assigned of from entire group within a backlog. The developer will pull the story and start creating it, start coding for the stories. Meanwhile, the testers are QA'ing the stories. So that is one way to utilize developers. Now the second way is, we used to definitely encounter a lot of bugs and they will be busy fixing the bugs. But most of the time developers are more occupied in order to work on the user stories, which are intended to deliver in upcoming sprints.

[12:09] RESEARCHER:
OK. So, they start in Sprint two already when you are testing?

[12:14] PARTICIPANT 29:
Yes, kind of.

[12:19] RESEARCHER:
I still believe there is an element of waterfall in this.

[12:24] PARTICIPANT 29:
So, I would still not say that's a waterfall because again in waterfall what you do, you complete an entire set of your release deliverable. And then you put an entire set to the testing. Here we are picking up the deliverable in small, small chunks and signing off those. So once a smaller feature is ready and you can put that in production but in waterfall, we put an entire set of features in the release.

[13:01] RESEARCHER:
OK. Let me give you another challenging question. What do you do with the dependencies across stories? For example, I have Story one and I have Story two in Sprint one and they are dependent of each other, but Story 2 was ready for testing but Story one is not ready yet for testing. So, what do you do?

[13:24] PARTICIPANT 29:
OK, say for example there are two stories. In one story, I am creating a UI/UX. I'm pulling the data on the expressive way. So, what we do, we keep a minimum acceptance criteria as part of sign off for one user story. We just used to quick check the UI/UX whether it is perfect or not. And also, data is ready, we used to give a bigger acceptance criteria to qualify the story.

Infinitely the first user story which is just to qualify UI/UX, the story points can be very less maybe one or two. Okay, but when we go for an entire set of integration tests, which is Story one and two testing entirely then those story points could be bigger, maybe three or five. So, there's one scenario. Now, let me give you one more scenario.

[14:24] PARTICIPANT 29:
Now, I face a couple of challenges where I cannot test my story entirely with XYZ factor. Maybe those XYZ factors could be a middle tier or could be our data. I cannot completely test without the same data on the page, right? So, what we do, we used to mark the story on only development, and we keep a way out from the story. Once we have it at our level, or once that dependency is resolved, we used to pull that story into the testing. And then test the story and then sign off.

[15:05] OK. I see. Alright, let's move to different questions. So, what do you think of Agile in general? What is your opinion about Agile?

[15:24] PARTICIPANT 29:
So, I have worked in Agile for seven years now. Agile is a methodology which will create a pressure on each other every individual to complete their deliverables. I'll tell you why.
So, for example, the PM has marked the story ready for development. It will create a pressure for developers automatically to complete development and mark the story ready to test. And now everything has to be done within a time frame where our entire team is agreed upon. Once they are locally signing off, it's pressure because I can't block any user story to get pulled away into the next sprint only because I was unable to QA. What I'll do maybe in stories ready to test, I have less time to complete QA, but still I'll put some extra efforts to complete the QA. We are kind of ready to deliver feature efficiently as we can. Like I said, mostly I am good with the monthly release where the sprint could be a pipe dream, but I saw people were literally releasing every two weeks, every ten days. And they are just keeping sprint on a weekly basis. After we done, they do staging and then release within ten days. So, by doing this, they are kind of introducing new features to production in a very particular way.

[17:24] RESEARCHER:
OK. So, you talked about creating pressure on the team? Is that a good thing?

[17:31] PARTICIPANT 29:
In the I.T field, I'd say sometimes it's good. All the resources are not the same. If you were lazy, there are a few. There were a couple of developers that weren't able to deliver efficiency within single. So, say for example, I had experience that when the developer team [inaudible] who

were developing five user stories. Now they had introduced one more developer who is kind of super-efficient. He was able to deliver all five user stories alone. I'd rather say maybe I had misled by putting pressure, but I would rather say it will create a more efficient. It will bring more efficiency from each of the individuals.

[18:35] RESEARCHER:
How it does that?

[18:40] PARTICIPANT 29:
So again, definitely then there is a PO who is standing up giving XYZ stories to the team. What is to be delivered into the sprint. He was doing a grooming meeting then he's doing a planning meeting. They can call out their blockers or concerns if XYZ story cannot be delivered. Once all the team members are agreed upon, their stories are then being pulled. So, everything is happening with the consent of all the team members. And then they realize there is more serious concern and their story cannot be marked as a draft.

[19:30] RESEARCHER:
OK. So, so let's move to some other questions. Can you describe the Agile setup of your work? What are the various rules? What is the process like? You can pick up one of your examples of one from your experiences and share that with me?

[19:58] PARTICIPANT 29:
We use Scrum. So, for one example, the Facebook search. The Facebook profile, what we see. Now the entire Facebook profile page cannot be delivered within a single sprint. So, what you do is draft a higher requirement that I want Facebook profile page. Page should have updates, which should have a search feature. Page should have integrate feature and so on. Now after having this draft, you will create small chunks of a user story as part of billiards. Now you will assign this user story to any team into the backlog. Now that is the difference between the sprint backlog and a team backlog. A team backlog is something which needs to be delivered within entire release and the sprint backlog is something needs to be delivered within a single sprint.

[20:06] RESEARCHER:
OK. So, let’s use the word Agile for this setup in the remaining of the discussion?

[20:09] PARTICIPANT 29:
Sure. Clear.

[21:09] PARTICIPANT 29:
Once this entire backlog has been assigned to any team, the Scrum master will work with the product owner very closely to see what our stories can be prioritized to pull into the current sprint. Once PO and Scrum master agree that there are five user stories, they want to deliver in sprint one, so those stories will be pulled into the sprint one. Then there will be a grooming meeting happening. Within the grooming meeting, the agenda is to bring every team member on the same page as part of the deliverable for the single user story. What different roles can be there in the community? Developers should be there, product manager should be there, product owner should be there, QA would be there, and the Scrum master would be there. All the team members are on the same page for the user story. They put the story points the into the planning. As a QA, I'll see this would be complexity is medium. I would rate it as a file user story point. And maybe I'll take 15 hours to complete my testing. Once the entire thing is done, the planning meeting is done, development will start working on these stories. Once the story is ready to test, we start picking up the stories and we sign off the stories.

[22:40] PARTICIPANT 29:
So, I'll summarize. The product owner is a crucial role. Scrum Master. Sometimes I see the product owner and Scrum master as the same person to remove the redundant roles. Then there is the developer, the project manager, and the QA. These are the roles I've seen so far with the Scrum.

[23:06] RESEARCHER:
OK. Who does the requirement in this team?

[23:10] PARTICIPANT 29:
So, the requirement is been given by product owner. PO.

[23:13] RESEARCHER:
And how is the QA engaged in the process?

[23:19] PARTICIPANT 29:
So, QA is actually engaged from the beginning. Like I said, those are part of the grooming meeting through to the retrospective meeting. The grooming meeting understands the acceptance criteria in all the deliverables. Then testing, then maybe by the retrospective meeting they can call out what went wrong to improve upon in the next sprints.

[23:51] RESEARCHER:
So, the QA is empowered from the beginning?

[23:57] PARTICIPANT 29:
Yes, and ideally, they should be.

[24:00] RESEARCHER:
What does it give them when they are engaged and empowered from the beginning?

[24:06] PARTICIPANT 29:
Sorry, can you say again?

[24:07] RESEARCHER:
What does it make them feel when they are engaged and empowered from the beginning?

[24:15] PARTICIPANT 29:
So, when the QA are empowered to join any activities from beginning, that will give more confidence to them to qualify the user stories as part of deliverables because they know what's going on with the user story. They know what has been discussed with the PO, they know what has been discussed with the developers. They know what to test. If they have any questions or concern, they can reach out to any of the members and clarify those concerns or questions and with the consent of all the team members, QA signs off.
What I mean by concerns, say for example that you use a user story and QA has missed one of the requirements. Now the requirement is bigger enough to pull into another new user story. So, QA already knows that this requirement, he has to pull as part of a new user story. So, this sort of things can be taken care if the QA are kind of empowered from the beginning.

[25:23] RESEARCHER:
Do you feel as a QA you are equal to the developer and the team and to all people in the team?

[25:33] PARTICIPANT 29:
I think as part of understanding of the user story, I will say yes. We all are on the same page. But everyone has a different set of skills. So, I cannot do as much good coding as a developer does and a developer can't do as much good testing as I do.

[25:54] RESEARCHER:
But emotionally and psychologically, do you feel like you are equal to the rest of the team?

[26:02] PARTICIPANT 29:
Yes. Yes, I do.

[26:03] RESEARCHER:
Yeah, so that's been given to you from the authority of the organization, right?

[26:13] PARTICIPANT 29:
Yeah, and that would relate to your situation that it is only because I was empowered from the beginning, I had more confidence.

[26:25] RESEARCHER:
This thing this team you described to me is co-located?

[26:31] PARTICIPANT 29:
This is co-located and I worked with the same structure where the team was not co-located. I was the only QA where the rest of the developers are from the US in effects.
But the same methodology is working fine with everything is co-located or not.

[26:52] RESEARCHER:
OK. Can you take me through the journey of a user story in this example and from the requirement of writing to the testing and the release and during this journey, can you explain to me, how do you assure quality?

[27:14] PARTICIPANT 29:
So, I said earlier Researcher, you will write out the user stories and as a product owner you would mark what they want to deliver as part of the user story and then mark the acceptance criteria as well. Now with me, when the story comes into grooming, what we discuss is the use cases. What are the different use cases possible to test this user story?
And once we pull out only the use cases within the grooming meeting, we should story point the user story. And while testing needs to kind of write out detailed test cases out of the use cases to qualify the user story. When I say pointing the user story, it's the consent of all the team members. So, for example, a single user story has can be a file pointer story, but from the QA it could be three pointers. So, what we do, we used to work together to finalize the story point and come with a single digit. We agree to a story point of five with the user story.

[28:36] RESEARCHER:
OK. So, let's move to the next question. What do you do to assure software quality in your Agile setup?

[28:57] PARTICIPANT 29:
For software quality in Agile setup.

[29:01] RESEARCHER:
So, in this Agile example you've been talking about, how do you assure software quality? How do you make sure that you deliver quality product?

[29:12] PARTICIPANT 29:
OK. So definitely during development cycle, we used to test all the acceptance criteria along with the regression. Now that you can use story, now at the end when all the stories has been completed. In the beginning of the next sprint Researcher, we used to take one task into the maintenance to test all the stories together, which we was delivered in the previous. And tested together with a very early start in the new sprint because QA has less work. They're under-utilized so make sure they are utilized completely. You should test two stories only. We do integration test and once all this sprint is completed, we used to keep one week of buffer before going to production where we test all the stories together in a staging environment and then sign off.

[30:22] RESEARCHER:
My understanding is there is a lot of focus on testing.

[30:28] PARTICIPANT 29:
Yes.

[30:29] RESEARCHER:
But testing is not quality in itself. Testing is quality control. Quality assurance is more than that. Quality assurance is doing inspections, is doing code review is reviewing requirements. As I said code review etc and continuous improvement of the process - all of that is quality assurance. So, testing only confirms that the product fits the purpose. So why in this environment, you don't have more than testing? Do you have more than testing? What do you do?

[31:18] PARTICIPANT 29:
Yes, so I mean we do not get involved into the planning and post assessment. So, let me give you example. The concept of definition of ready that's been introduced from QA's end. Why? Because earlier we used to face a lot of difficulties as part of getting clear requirements and what part of this requirement had changed. We had escalated this in the grooming meeting and then we came across this definition of ready, where we confirm that once the user story is

marked definition of ready, we cannot change any acceptance criteria. So that's the one of the processes we created. Then there's a lot more processes we created, like how we want to carry out the testing in staging environment. What all developers should do when they want to mark a user story ready for test. So, what we do, we create a basic set of test cases. We used to provide that to the developer and developer will quickly run through those test cases to make sure that things are basically good as part of the user story. So, you know definitely QA is involved into creating a different setup process, needs functional, not functional.

[32:47] RESEARCHER:
You mentioned that you don't change any acceptance criteria after a certain stage in the process. Isn't that against Agile spirit with which comes change?

[33:08] PARTICIPANT 29:
No, I wouldn't say because Researcher. So, for example once use story's point, the developer has start working on it. Now maybe a week or ten days, QA is coming back saying, I don't want this story is to be created in this way. I want this feature is created in a totally different way. Those ten days have been kind of wasted for the developer. Maybe some of this has been wasted from the QA's end or the product manager's end. So what we are encouraging for the PO is once you confirm the deliverables, once you confirm the stories that OK, you want this one and then put it into the story and assign it to us, instead of frequently changing the acceptance criteria. There's no end to it because say for example, I am testing this story. I am coming up with the 50 new scenarios that cannot be part of the single user story. So again, the scope of the story will be increasing, no matter what we had signed off. Just to ensure there's no unwanted surprise. We would rather stick to the story acceptance criteria, what we had sign off earlier. So at least with the delivery we achieve the story points or velocity what we had signed off earlier.

[34:44] RESEARCHER:
How about if the business requirement has changed at that stage?

[34:49] PARTICIPANT 29
So, we saw that issue earlier when you delete a page, and when you are in the middle of computing that page, that page what was actually discarded. People don't want that page to be created at all. They approach when to scratch. But again, that will be a business decision and nothing to do with QA. So now if QA is coming up with requirement in the middle, this is saying that they don't want this thing to be XYZ way, right? I think we cannot control it. We had to kind of step the walk and maybe we need to create a different user story and start working on that.

[35:38] RESEARCHER:
So, do you think the change is an external factor that is beyond the scrum team itself?

[35:50] PARTICIPANT 29:
No, I didn't get that.

[35:51] RESEARCHER:
So, do you think the change is an external thing that it doesn't matter the delivery of the team. So, you keep delivering even though the change mess up your project, right?

[36:06] PARTICIPANT 29:
No. So definitely that we have retrospective meeting to call out these instances which will give us some space to grow upon. So, say for example, one instance where the requirement has been changed. Maybe we kind of entertain that requirement, but after having a retro meeting, the team will see what went wrong while drafting the user story. And the improvement area is what can be done to manage last-minute surprises and come out from that situation. Because every team has things to improve, I would say. In Agile every person can do better with what they delivered in the past sprint. So definitely there will be huge learning.

[37:01] RESEARCHER:
What are the challenges of a QA team in Agile in general?

[37:08] PARTICIPANT 29:
Definitely one challenge I would call out is we used to receive our stories very late in the game. That creates a lot of pressure for us to complete the user story within a given timeline. So, say for example, there is a one-week-old sprint. They would give us a story to test by Friday. And that will create a lot of chaos to qualify the stories. Another thing I would like to call out is change in requirement. Once we started testing, once we raised questions, they say okay now I want to check the requirement. So that's another challenge but most of the people, most of the time we, the stories are being changed for testing is the most challenging thing for QA.

[38:14] RESEARCHER:
Okay, let's move to the next phase of the interview. Alright. This is a little bit of a provocative question. Do you think the Agile setup, the example you shared with us during this interview, do you think this Agile setup produce quality software? And how?

[38:50] PARTICIPANT 29:
I would say yes. It does deliver quality software. Everyone in the market are using agile methodology. Well, I'll say most companies are using Agile. I've seen companies have migrated from a waterfall project and I have seen many. Why I say quality product because we are testing one component within a sprint. We would do integration tests in given days. Now once all the changes have been done, we used to keep one week of buffer where we stage this code together and test completely. Within this week, even project managers, product owners, UATs are kind of working together in order to qualify all the stories. What they've already done into the sprint basis as well, which will give more confidence to the stakeholder that this XYZ thing has been signed off by multiple people, not for the single time but multiple times. And definitely that will give you a better experience to customize them.

[40:24] RESEARCHER:
Can you share with me a positive story from your Agile experience?

[40:31] PARTICIPANT 29:
Sorry?

[40:32] RESEARCHER:
Can you share with me a positive story from your Agile experience?

[40:38] PARTICIPANT 29:
A positive story? I didn't get that Researcher.

[40:40] RESEARCHER:
Something good happened. You delivered a good feature that didn't break the system. I don't know. Something really good happened. Yeah from who?

[40:57] PARTICIPANT 29
I think the users are to draft a better requirement. For example, me writing a one user story and you writing one user story, you can do better writing so that your item can be easily understood by all the team members. We used to help the product owner in order to write up the requirement in an easy way so that every team member can understand very quickly. We created a couple of processes which I just said definition of value, which is been like for every folks. And the company decided to implement that company-wide for all the locations. We did create one process for testing early days into the sprint for the last sprint deliverables where the QA is underutilized. By doing this, we are utilizing the QA as well. We are kind of achieving higher quality standards because we are testing it again in the early days. So those sorts of

things, I think went well. One thing I would like to call out is the rapport of the QA among all the team members is very important.

[42:15] RESEARCHER?
The what? Sorry, I didn't get that word?

[42:19] PARTICIPANT 29?
Sorry?

[42:20] RESEARCHER:
I didn't get the last word.

[42:23] PARTICIPANT 29:
Yeah, so your rapport.

[42:26] RESEARCHER:
Ah, your rapport.

[42:28] PARTICIPANT 29:
Among the team is very important with each other. The team members have to do good work when they work with each other. Because say for example, I think we won't help developers. If I had a suspicion if I reach out to developers, they won't answer me. I think that would create a negative one. So, it's team collaboration and coordination is the topmost thing within Agile.

[43:03] RESEARCHER:
Yes, correct. So how do you build the reputation within an agile team?

[43:09] PARTICIPANT 29:
So, I think Researcher, we are being more communicative by grooming and planning meetings. Reaching out to developers frequently. If I have questions and the developer has something for me to answer, I can answer that as quickly as I can and not pulling them back. So, say for example you reach out to me with a question and if I'm answering you the day after, you might not feel good. But if I answer you within 15-20 minutes or one hour you feel good, right?

[43:45] RESEARCHER:
Yeah, correct.

[43:48] PARTICIPANT 29:
So those sorts of things can be best practices as part of building a rapport. And topmost thing is your knowledge. How could you understand the thing, how you deliver the thing? That will definitely compliment your rapport.

[44:04] RESEARCHER:
Fantastic. Now, can you share with me a negative story? Not everything is perfect in Agile worlds, so there are things that happen.

[44:19] PARTICIPANT 29:
Yeah so, limited communication. There are a couple of instances with communication with a Scrum master who leads the communication within a development team and that communication is not shared with QA. Because of this, we used to miss out couple of scenarios while testing. This was a big concern that we raised. So, every team member should be looped in into any communication regarding the deliverables. That is the one thing. Another thing is changing requirement that was happening very, very frequently when I joined this company in the beginning of my career. And that was kind of messing up everything because developers were not enjoying creating the code because the middle of something, then there are a few comments, like okay do it way. And they had created a structure in another way. On the QA side, we are done with the test case plan and creation and everything. Then the PO comes up and says okay, do it another way. I had to rework everything and that creates demotivation.

[45:41] PARTICIPANT 29:
So, we worked a lot to get it fixed and third thing is stories are being done in the late phase of the sprint. So those are the three negative things I had seen so far. I did face couple of issues pertaining to the attitude as well. Maybe a bad attitude of scrum master or the developer because all the personal things but eventually we get it fixed. It's a team we had to work together as a family.

[46:19] RESEARCHER:
That's good. What is the relationship like between the developer and the QA in a Scrum team?

[46:39] PARTICIPANT 29:
So yeah, I would say a mature QA and developer. I'm a mature QA and I'll be more concerned about quality and if I'm a mature developer, then I'll be more concerned with the code quality of what I'm writing. For example, if I'm a mature developer, I'll be seeking more good testing scenarios. But yeah, I saw some developers who are very early into that career and they get offended the bug being documented.

[47:13] RESEARCHER:
Yes, correct.

[47:15] PARTICIPANT 29:
But eventually we had to help them understand we not doing something personally, it is just that we want to deliver better quality to our customer and that's the way we found this bug. It it's reproduceable then we have to fix it. And it's those two things. Now the quantity of mature developers are being increased in the market.

[47:47] RESEARCHER:
How is that happening?

[47:50] PARTICIPANT 29:
I think people understanding the deliverable. Nobody used to like bugs made in the team. Maybe if somebody found a bug in the staging or production that would be a bad impact not on the QA, but on the developer as well. Because it just that bad quality of code has resulted in this. It's the entire team who missed this. As QA, I'll try to code as much scenarios as I can being a mature developer. I would try to seek as much scenarios as I can from the QA side. And you and I going and unit test those. I would like to highlight one thing. A couple of developers have created a code review process. Say for example, they created a code and they give that code to another developer to make sure the code quality is good. The scenarios have been covered and it's not qualifying the entire thing.

[49:07] RESEARCHER:
How does this agile setup you using in your team help achieving software quality?

[49:17] PARTICIPANT 29:
Ok, I'll summarize. Scrum makes the team more efficient. Efficient team produces better product and software quality. Most importantly we are empowered; this facilitate sharing information with better transparency. Bugs are found and communicated without fear or feeling of guilt. The team operate better, because our relationships are strong. We feel very close to each other's and minimum power over us. We invest more effort on quality and we produce code with less bugs.

[49:37] RESEARCHER:
Fantastic. Now, we move to our final question and it's a little bit provocative just to create a bit of discussion. What do you think of this statement: Agile produces poor software?

[49:47] PARTICIPANT 29:
If the team is not following a process in Scrum, then definitely the quality of the software will be poor. Okay. Let me give you an example. The team is not focusing on acceptance criteria. The team will not focus on the deliverable within the timeline. The team will not focus to complete testing within time. That's definitely leads to poor quality and that will create a lot of [inaudible]. Eventually at the end what you will see, what I want to complete a hundred stories with the three sprints. Now at the end, I can see only ten stories are completed and ninety stories are still there only because the developer hasn't completed their work in the timeframe. Or the PO has not completed the requirement properly. Scripting will only give you better quality. If not, then agile will give you the worst product because eventually people will end up conflicting the customer as well because they were meeting the timeline as well.

[50:58] RESEARCHER:
So, the quality of the implementation of Agile itself is very important to deliver software quality. That's what you say?

[51:08] PARTICIPANT 29:
Researcher, I would say not in Agile. It has to be software development cycle. You have to follow process.

[51:15] RESEARCHER:
Yes. You have to follow process.

[51:16] PARTICIPANT 29:
By following process, you deliver better quality products.

[51:20] RESEARCHER:
Yeah, correct.

[51:22] PARTICIPANT 29:
In any methodology you have to adhere to process. That's the Silver Bullet.

[51:35] RESEARCHER:
Yes, I agree with you. That's well said. Do you have any other comment or thought you'd like to share with us before we finish the interview?

[51:49] PARTICIPANT 29:
I think giving a respect in Agile is something you know; you respect your team or your peers. Respect the developer, PM, or product owner or QA will definitely produce a good rapport and that good rapport would definitely help to deliver a better-quality product. Make sure you are asking each and every member whether maybe within a grooming meeting, just call out as a product owner that QA and developers, are you all good with these stories. While pointing the story, call out their name and you do a little boost into their confidence. It will eventually help to get their better work.

[52:39] RESEARCHER:
Fantastic. I'd like to thank you PARTICIPANT 29 a lot for your time and for your valuable input and your knowledge. You seem very passionate and knowledgeable of your stuff. Good on you. Congratulations. I really like to talk to people like yourself with passion for what they do and with in-depth knowledge of their stuff. Yeah, it's not quite often you meet someone passionate and knowledgeable of their stuff. So, thank you very much. Do you have any questions for me?

[53:20] PARTICIPANT 29:
Nope, Researcher. I'm kind of happy with the interview. Most of the time what I see in interviews, the person who's taking interviews are talking more instead of the person is giving interviews.

[53:54] RESEARCHER:
No, I shouldn't be talking more, I should be talking less.

[54:00] PARTICIPANT 29:
You have done it perfectly. I really enjoyed this kind of interview. Just let me know if you need anything from my side and I'm happy to help you.

[54:12] RESEARCHER:
What we usually do is a quality assurance thing. We transcribe the interview and I'll send it to you by email. Just check and have a look whether everything is okay; we haven't misquoted you or anything like that. And that's it. I would appreciate.

[54:35] PARTICIPANT 29:
Good. Thank you.

[54:38] RESEARCHER:
All right. Yes. Thank you very much. Have a good day.