PART 2:

00:01 PARTICIPANT 39:

So, will not allow any attack. I mean, the hackers are continuously trying to attack you, for tests or sites so to prevent both kinds of attacks. And the other one is like, there is also a concept of DDoS attack. So, it's like the attackers who hit your service endpoints, or your software endpoints, I mean, the URLs, multiple times sort of particular timing stems, so to break down your systems, if it is not capable of handling that request, within that instance. So, for that, in order to facilitate to prevent those are DDoS attacks, we also carry out the performance testing of the software we deliver. So, during the performance testing, we test for the load I mean to say the number of requests per second, the particular endpoint of my service, or my application can handle. So, in that, in that way, we get our threshold or threshold of the requests per second. So, once we will fix that number of requests in case of DDoS attack also, there are mechanisms for use like circuit breaker concept, to allow the number of requests to our infrastructure, or our software, it will be blocked by the software itself. And it happens in the network layer. It's like we have the concept of API gateway. And API gateway itself provides the cross-cutting concerns like we maintain our operations and send content delivery network. That is the CDN like all our static objects, like images and texts, those are displayed constantly throughout the webpages, those are maintained in the CDN layer. And these are all in maintained in the API gateway layer. And these are DDoS, attack and are insert in that IP two layer. So, this is how we ensure the software quality through this performance testing and security scanning all the things and mostly quality. If it is quality, then definitely it's a testing team's effort, performance testing and automation testing. And also, manual testing. So, all these terms as a whole define the quality of the software.

03:20 RESEARCHER:

So, at what stage the end user does testing?

03:28 PARTICIPANT 39:

Once our software is ready, and we will deploy it on. First, we follow separate enrollments for testing then UAT is prior to the production. So, once we'll deploy our software to the UAT then we provided for us the third party and user to test our service or our application.

04:01 RESEARCHER:

Okay, is the end user testing done by the QA or the user themselves?

04:07 PARTICIPANT 39:

No. QA testing is done in the testing environment. And we promote our code from dev environment to test environment, then QA testing happens. Once all the issues are fixed and QA clears it from the queue, then the product owner will go for the UAT deployment and then the product owner communicates with the end user or the business team. So, for testing of the application.

04:41 RESEARCHER:

So, what's you have been describing so far, they are quality assurance processes and tools, which they have existed before Scrum or Agile. So, I have a challenging question for you. What has Scrum brought new to the picture to help achieving software quality?

05:10 PARTICIPANT 39:

So, the main thing is like during the testing like previously what was happening like compared to the waterfall model, because previously we were working only or following only the waterfall model. So, in waterfall model what happens it', I think, I have to answer it in other way. Like we have a product and whatever product we have that is part of whole product, but in waterfall model, you can consider a product as a whole. So, the team size will be very big in that sense. But in Agile, there are multiple teams and later on all these teams effort are getting integrated to make a product as a whole. So, if we face any challenge, and we have any dependency on other teams, so that can be well informed or communicated to the team so that is the benefit we get. Like since they are a separate team that will split up means the delivery will be split up since they are concentrating only on that part of the software product. But if we go compare it to the waterfall, since we have consider everything as a whole then even though some other team is involved in that module, but their time to market will increase. So that is definitely a communication problem for time to market to increase in case of waterfall. That's compared to Scrum.

07:27 RESEARCHER:
Okay, I will follow up these questions. My understanding so far from your answer to what Scrum or Agile brings to the software quality, it's small deliverables. So, the team can focus in small and concise deliverable. The communication between the team and you mentioned time to market. So how does small deliverable and the Scrum process help software quality?

08:03 PARTICIPANT 39:

Okay. You software quality?

08:09 RESEARCHER:

Sorry?

08:10 PARTICIPANT 39:

It's no time to market. It will speed up our delivery process or the product delivery timeline. But software quality we answer all the steps we already discussed.

08:26 RESEARCHER:

Yeah. But my question to you what Scrum brought new to the picture to help software quality. So, you talked about small deliverable, communication within the team, how this quality helped software quality, how these qualities or advantages of Scrum help delivering software quality?

08:59 PARTICIPANT 39:

So, quality. You mean the product quality?

09:10 RESEARCHER:

Yes. Yeah, the product quality. Yes. So, for example, does communication help solving problem and therefore better quality? Or does time to market means that the end user can test more frequently? So, how these qualities of Scrum helped to achieve quality?

09:44 PARTICIPANT 39:

Yes, so, since we follow the daily Scrum meeting, so everyone is involved in that Scrum team from product owner and business analyst and development team or testing team. If anyone needs talk at any point of time, so I think we already discussed this one. So, it's like, if within our team, there won't be any blockers, if there is any blocker then we need to resolve them at the very beginning. So, it means everyone expects to involve or to clear that blocker as soon as possible. So, it will speed up our development effort since everyone is involved in, it's not like a single person's task. So, your other team so, from the requirement perspective, if we have any kind of doubt, then the business analyst and the product owner they will clear our doubts. The other thing is if there is any changes in the requirement during our development, then the product owner will inform it through our business analyst. So, we can make changes in our requirement. So that once our product will be delivered or at a later stage of time will not get the requirement again and we need to do the rework. So, that is the benefit, it also at the end it also adds to the quality of the product quality in that sense, it is as per the expectation of the end user of the business team. So that is the benefit we get out of here.

11:40 RESEARCHER:

So, my understanding is this, Scrum helps better understand the requirement, therefore, you meet the end user expectation from the product, correct?

11:57 PARTICIPANT 39:

It's correct. It's like in the it can achieve the real time goal like during your development process, if you will get any changes in the requirement, we can capture that requirement in that sprint itself or you can keep it in the backlog and duly considered in the next sprint. So, to the spirit of the delivery as alerts quality or quality in the sense the expectation of the customer. So that the end product don't have that feature, which is not expected by the customer. This is because we have already considered their requirement in the during the development phase. And we make the changes accordingly.

12:56 RESEARCHER:

Okay, that's the end user expectations. So, for the code quality and the design, how does Scrum help?

13:07 PARTICIPANT 39:

Code quality and for good qualities like we follow a standard code quality practice among our team. So, there are certain approaches. And we follow the code review practice also and it is the peer review like anyone on our team reviews the code before committing to the source code or the repository. So that into the code quality and there are also use this static code analysis tools in our ID basically currently we're using this Eclipse and IntelliJ. So, these are more of a technical perspective.

14:11 RESEARCHER:

I do have another challenging questions, code review and static analysis can work in waterfall, there is no problem to use these quality assurance technique and processes in waterfall. What makes them better in Scrum?

14:34 PARTICIPANT 39:

So, these like it's not like something Scrum. Scrum is just like a process like waterfall. Or better in the sense like since we have better process to code. And do we have multiple Scrum teams for the same product. It means at the end; the code quality will be better if you compare it with the preferred model where the team size is very large. And it's quite, not quite, but it may be a bit challenging to ensure the code quality within a large number of team. So, if we consider Scrum, we have a small number of teams and our development scope is as a part of our product as a means it will consider product as a whole, then we are involved in maybe few modules. One thing is, we'll deliver maybe a couple of modules, three, four modules, if it has several modules, so it's better to maintain. We can better maintain the source code quality, if our source code itself, the number of lines is less or the number of classes we write is less. So, in that way, it is manageable like this. In our Scrum team, we will ensure more code quality as compared to a larger team.

16:17 RESEARCHER:

Okay, so what I'm hearing is the size of the team make it better or enhance the ability of the team to write better code. So, because you're close to each other, you communicate better in a small team?

16:38 PARTICIPANT 39:

Yes. And other thing is like in our Scrum team, there are developers or testers who are having more experience and there are also developers and testers are having less experience. So, since our team size is less so these, the experienced people can help the less experienced people they can groom. So that they can also achieve a speed in the development following the Standard Code practice and maintaining the code quality. So, this is the benefit we get. But in waterfall since it's a very large team. So, it may be a bit challenging to groom everyone in the team and maintain the code quality and speed up the process. So, this is the major difference I can see from my own experience.

17:41 RESEARCHER:

Okay, I'll move to some questions. And I need examples here. If possible. How does for example, Scrum in your team helps finding bugs?

17:57 PARTICIPANT 39:

Finding bugs. It's mainly through our testing team. At first, they do the manual testing and also write the automated test scripts and the requirements. And from there, from there only we come to know our bugs. So, if required, once we complete our delivery, for software delivery, once we complete our deployment phase, then we'll move to the deploy our code to the test environment and then they are itself the testing team gets involved at the start the testing of those software, both manual and automatic testing. And once the automation scripts are ready and will that script itself executes during the deployment also. So, we can come to more defects also, where the test case will fail during deployment. So that is our common process.

19:08 RESEARCHER:

So, this process existed before Scrum in waterfall, you could do this, how does Scrum help this process?

19:21 PARTICIPANT 39:

Scrum is basically we discuss like it's a specific module of the application and it involves lesser number of team members. In that perspective, it helps the quality better, quality of the delivery.

19:41 RESEARCHER:

So, in what way... Yeah, yeah?

19:43 PARTICIPANT 39:

One more thing I think I missed is like, you need to think of communications. We have already discussed it likely if we have any dependency on other team, then we can make them our [inaudible]. They will prioritize our requirement in the Scrums or their daily Scrum so that we can get our requirement for dependency to be resolved as soon as possible so that it will not block our development phase. So that it communication well in advance you can do among other things. Because on lots of tests, all these modules will be integrated to make product as a whole. But all these things in waterfall we have to plan from the beginning and if will get any change in the requirement, then we have to reconsider our architecture and we are don't know what all modules are we effect affected by this. So, less on the delivery timeline, in case of waterfall.

21:09 RESEARCHER:

So, in Scrum, the intra group communications help, for example, better communicating defect and accepting errors, and hence it helps quality if that's what you're saying?

21:25 PARTICIPANT 39:

Yes, mostly our dependencies are resolved before we start our work, or maybe after we finish our work, they are supposed to, or they already have the dependencies so that they started working on that dependency to resolve that dependency. And so that took a lot to block our development. So, for example, let's say we are working to create one application, which required to call another endpoint that brings some data from the legacy system. Let's assume in my last two that we are fetching some data from this app. So, for that, the provide us one endpoint through which we call their service, and we get the data. So, let us imagine if that endpoint is not ready, then we are blocked in that case. So, we inform well in advance during our sprint planning, when we discuss our architecture that we need this endpoint to be exposed, so that we can face this data. So, we informed them in advance so that it will not block will not be a blocker. So that is another advantage. I can see.

22:53 RESEARCHER:

Okay, so the next questions, you talk a little bit about it, but I will still ask it anyway. How does for example, if you can give me example, Scrum in your team helps produce in high quality code?

23:11 PARTICIPANT 39:

In our team, we strictly follow, not in the current team, all my previous projects will strictly follow the code review. And the code review process helps to maintain the coding standard, as per the Java specification or the latest Java standard. Since all my projects are in Java,

mostly currently in [inaudible] microservices, we follow the proper architecture like we should not go for any implementation of any anti patterns. The few complex projects also I have worked like those are involved in [inaudible] bash trimmings, all the stops, so and mostly those customers were like their internal team, militarized [inaudible]. So, these [inaudible], so all these tests lies in the demilitarized zones are not exposed, only the API endpoints are getting exposed. So, it ensures the security. In terms of code quality, it's like before, along with this code review, we also follow the code analysis tools.

24:45 RESEARCHER:

So, code review and code analysis tools.

24:57 PARTICIPANT 39:

Yes, mostly we follow the reports are published. The default threshold is set as eight percent, so below eighty percent is not allowed to be deployed.

25:11 RESEARCHER:

Yeah. But these processes again existed before Scrum. What's the difference? Or what is the value that Scrum brought to code review for example?

25:28 PARTICIPANT 39:

Scrum before Scrum is like only, we have heard the waterfall model. I have worked in waterfall and waterfall we already discuss is the product itself we have to deliver as a whole but in Scrum, it's not like that. We have different modules we have created or separated the product from different modules with other different modules. So, since our code bases less than the coding process will be more effective. Can repeat your question again?

26:08 RESEARCHER:

Yeah. You answering it's quite good. So, you think that Scrum brings efficiency, right?

26:15 PARTICIPANT 39:

Yes, definitely.

26:17 RESEARCHER:

Yeah. So how does efficiency of the process help delivering code quality?

26:26 PARTICIPANT 39:

Mostly this even though all these technologies existed before waterfall also, but in that waterfall model, you have a large number of code base, let us assume you got a number of issues, and your application is not getting deployed. And the team as a whole need to fix the issues. So, if one module is not having any issues, but since the other modules are having issues, so the developer who has returned that piece of code, he needs to fix it so that it can be delivered. But in our case, even though one module is ready and deployed, the other module is not ready. But we can go for the testing of it. Certainly, necessary delivery process once that the other our modules, developers will fix the issues, they will complete their

development. So, in that way, like the development process will be faster. It's not dependent on other modules, but if you follow the waterfall, then we have to fix the project as a whole. Everything you have to fix before and challenge the major challenges if we get any requirement change. We keep in mind about the involvement of all the modules and what will be there after effect, if will make the changes in one module.

28:17 RESEARCHER:
Okay, fantastic. So great. That's a good perspective. Thank you. The next question is how does your Scrum setup or Scrum environment motivate you as a software developer to write or to achieve code quality?

28:43 PARTICIPANT 39:
Yes, I think that is a very good question. It's like previously in waterfall we had the requirement document, technical design document. Those are multi page documents. And it's not that easy to refer them every time you make any changes in your requirement or follow that before going for your development. But here what happens is currently I'm using JIRA Software, where we create our user stories, we create our epic based on the requirement from the product owner. Then we create our user stories where we mention our acceptance criteria and the details of the stories. Later on, we break the stories into different tasks or tasks into subtasks based on our technical requirements. So, you can think we break stories in very minute level. So, the tasks will be divided or created. The four different tasks will be created for testing team and the development team, so to speak. Separate the requirement, it [inaudible] and it will be easier like we also attach all our requirement documents or any reference document only during development, everything will keep it on JIRA so that it will be there for us during development phase itself, or during testing phase, the testing team will ask for any standard data for testing. Like we provide our endpoints and the request parameters or request module that are tested them test the endpoints. So, we provide the input for testing to the testing team also. And we attach all those required things in the JIRA itself. So, it's like better manage availability and maintainability I can say. So instead of going to check the technical design document, architecture document, every time again and again for implementation. And directly sending the documents to the testing team or keeping it in a certain location and the testing team we go to that location again and again to retrieve all those are things, and they have to categorize themselves that this is the record document, but if you will attach them to the specific user stories, then it is very specific to that implementation. So is the major advantage of using this Rally and JIRA Software and Scrum practice. Like we get the reports also, like our sprints, alerts, burn-downs, all those things.

32:01 RESEARCHER:

So, my understanding is the Scrum process, it's chunked the work into small deliverables, and it's create transparency on those deliverables. And it's motivates you to write better code. Is that what you're saying?

32:20 PARTICIPANT 39:

Yes.

32:22 RESEARCHER:

Can you share with me a concrete example of that?

32:30 PARTICIPANT 39:

Example you meant to say from my past experience?

32:35 RESEARCHER:

Yes.

32:40 PARTICIPANT 39:

Yeah, like I told in the waterfall model, he was working as a technical architect in that time at that project and it was for some banking from [Deleted to preserve the participant anonymity] So, the project here, we have to create one major reporting of the end product. So, you have to design everything well in advance. And even though we get a new requirements in this, we have to make regular changes in our code. And prior to that, also, I was working in [inaudible], it's a market research company. So, they were following this waterfall model, that they regularly get the requirements and regularly changing the code base. And later on, or during integration, we face a lot of challenges. Since the impact analysis is to be done willingly at work, but once we make our change, then only we will go for that we consider or take if any other module is broken or not. In the last project or in the current project we follow this Scrum team structure. So, there it's like we get our requirements even though we get requirements from the client later on also. So, these are not all that much major changes, and it doesn't have impacts on other modules also. Even though it has impacts but we can well in advance inform the other team to make the corresponding changes. So that the end product will be more efficient.

34:52 RESEARCHER:

Okay.

34:54 PARTICIPANT 39:

But I mean to say, if we are making something and just during the development phase, we can't know that it is having some impact on the other module, then we can inform them that you have to also make these changes based on this requirement change.

35:14 RESEARCHER:

So, the Scrum process gives you have more control over the code.

35:21 PARTICIPANT 39:

Yes.

35:22 RESEARCHER:

Yeah. So, you always change your code to better quality, correct?

35:30 PARTICIPANT 39:
Yeah. It's not always change. It's based on the change in the requirement. But do we ensure the better quality always ensure the better code quality.

35:42 RESEARCHER:

Okay, great. I'm just going to look at the question. I didn't ask because the conversation has been shifting. Can you share with me a positive story about how Scrum helped you to produce software quality from your own experience?

36:07 PARTICIPANT 39:

I think we repeating the same things.

36:10 RESEARCHER:

Yeah, we're repeating ourselves. Yeah. Yes, no problem. Do you have a negative story where a Scrum didn't help produce in better quality?

36:23 PARTICIPANT 39:

No, not yet. From my experience, I think the very first project when I was involved in the Scrum was [Deleted to preserve the participant anonymity], and I was quite amazed on the delivery part, like everything was already planned and everything was going very smooth, smoothly. So, like the Scrum master and the product owner, the way they have planned things under deliveries, timelines. So, everything were very smooth.

36:57 RESEARCHER:

Okay. The last question, we have two minutes left, or three minutes left. The last question is, what do you think of this statement, Agile produces poor quality software?

37:17 PARTICIPANT 39:

Yes, it is not true. It doesn't produce poor quality software.

37:25 RESEARCHER:

It doesn't, right?

37:26 PARTICIPANT 39:

Yes, it doesn't.

37:28 RESEARCHER:

Okay.

37:29 PARTICIPANT 39:

We're discussing that like, within others also, we get involved with our DevOps teams and platform teams. So, platform team will ensure these all these security, and platform related platform, I mean, the cloud or technologies of the cloud, where we deploy our code that is our platform team. And DevOps is our Jenkins pipeline team. They will create automated scripts that will deploy our code automatically once we make any commit to the source repository. So, DevOps team will integrate all these [inaudible] for code quality and security vulnerability checking. The platform team will leverage our these inter service communications and calling the external endpoints and setting the firewall policies. If we are calling any legacy or we are getting fetching data from any legacy system, they will deliver us to open the firewall policy as per the company advisory guidelines. So, this is a collaborative effort. And the most benefit is that lessen the delivery timeline because you directly connect to the platform team and the DevOps team. And we can also ask them to prioritize our work with our dependency. So that is a benefit we get here.

39:21 RESEARCHER:

So, this flexibility across teams facilitate also the efficiency of the process. Okay. Great, fantastic. Do you have any questions for me before we conclude?

39:36 PARTICIPANT 39:

So, what I want to know is, you are working on a research paper, right?

39:43 RESEARCHER:

Yes. It's a research project. Yeah.

39:58 PARTICIPANT 39:

My last project, I was doing work with Maersk.

40:07 RESEARCHER:

It's a Danish company?

40:09 PARTICIPANT 39:

Yes. Maersk is the shipping company.

40:13 RESEARCHER:

Ah, yes, yes, yes. It's the biggest company, Maersk. Yeah, it's the biggest company in Denmark. It's a national pride. It's global. What was the project about?

40:45 PARTICIPANT 39:

I was working at Maersk and that was the last project where mostly the platform team was efficient.

40:45 RESEARCHER:

Yes, the Danish working environment is very cordial, very friendly.

40:49 PARTICIPANT 39:

Yes, yes.

40:51 RESEARCHER:

So, you were located in Denmark or from India?

40:53 PARTICIPANT 39:

No, I was in UK.

40:56 RESEARCHER:

Ah, okay, you were in the UK at the time.

41:02 PARTICIPANT 39:

Yes. Yeah okay. It's fine, it was very nice to talk to you.

41:09 RESEARCHER:

Thank you very much. I appreciate your participation. And in the future, if you have any questions or you want to get in touch, please do.

41:18 PARTICIPANT 39:

Yeah, I think was able to answer all your questions.

41:24 RESEARCHER:

Yes. The conversation was really great. I appreciate, thank you very much. It was very to the point. I learned a lot from you. Thank you very much.

41:36 PARTICIPANT 39:

Like, it's whatever I explained is based on my own experience.

41:39 RESEARCHER:

Yeah. It's all we need, your personal experience. What we need is for you to share with us your personal experience, that's all we wanted. You did it very well. Thank you very much.

41:51 PARTICIPANT 39:

Thank you.

41:54 RESEARCHER:
Okay, I wish you a good day.

41:56 PARTICIPANT 39:
Yeah, same to you.

41:58 RESEARCHER:
Okay, bye from Copenhagen. Bye.

41:59 PARTICIPANT 39:
Bye.

42:01 RESEARCHER:
Bye.