[00:49] PARTICIPANT 12:

Morning.

[00:50] RESEARCHER:

Good morning, how are you?

[00:52] PARTICIPANT 12:

I'm good. Can you hear me?

[00:53] RESEARCHER:

Yes, I can hear you. Can you hear me?

[00:55] PARTICIPANT 12:

Wonderful. Yes, I can. Excellent.

[00:58] RESEARCHER:

Yeah. I'm really sorry about earlier. I don't know what's happening in the Internet's, why it kept giving us the wrong time.

[01:04] PARTICIPANT 12:

I don't know. I apologize. I wasted a few minutes of your time.

[01:08] RESEARCHER:

No, no, it's OK. It's a misunderstanding, which none of us is responsible for or as with Agile, not accountable for it.

[01:20] PARTICIPANT 12:

Great, no worries. It's all fine.

[01:23] RESEARCHER:

Do you have any questions for me before we start?

[01:26] PARTICIPANT 12:

I'm sorry?

[01:27] RESEARCHER:

Do you have any questions for me before we start?

[01:30] PARTICIPANT 12:

Just one. Do you mind if I share my screen?

[01:33] RESEARCHER:

No, no, not at all.

[01:38] PARTICIPANT 12:

I will be able to cross-reference images, or anything online based on the discussions we have.

[02:19] PARTICIPANT 12:

Can you see the screen?

[02:20] RESEARCHER:

Yeah, I can.

[02:22] PARTICIPANT 12:

OK. Excellent. So ready when you are.

[02:27] RESEARCHER:

Ok. Let's start with some introductions. Can you introduce yourself and talk a little bit about your experience?

[02:35] PARTICIPANT 12:

Of course. My name is PARTICIPANT 12. I've been working for the past sixteen years, to be more precise, in the information technology. And in particular, I spent the last ten years working as a software developer, project manager, and Scrum master. Over the years, I've worked in multiple industries, travel, engineering, online gaming, and gambling, Fast moving consumer goods. Yeah. In a nutshell, that's me.

[03:16] RESEARCHER:

Ok, that's sufficient. Let's move to the important stuff. Let's start with, what do you think of Agile? What's your opinion of Agile?

[03:26] PARTICIPANT 12:

I think it is both a buzzword and sometimes even underestimated or overestimated. If we think about it, being Agile isn't really new. Roman legions were Agile two thousand years ago. That's how they conquered most of Europe, Northern Africa, and Western Asia by working in small independent teams made of all specialties that were required in order to accomplish different goals. So, within the legions they had everything they needed, including engineers that were able to build bridges or, erect structure and so forth. So, they were quite

successful because of that. Now, when it comes to computer engineering, the term appeared around 1970s, and that's when a number of engineers decided that the traditional way of working wasn't quite right. So, they decided to make it more flexible. So that's why I think they went for the word Agile. So, in essence, I think, it's a term, it's a notion that has to be understood, way better than it is nowadays in many places around the world.

[05:02] RESEARCHER:

What is missing that people don't understand that well.

[05:07] PARTICIPANT 12:

What is missing? I think some people have mandates from, let's say from large corporations in particular. They have mandates from their managers, from executives and they are being told, OK, let's become more Agile. But the notion itself isn't really a noun. It is more an objective. So rather than let's be Agile, the better perspective should be, let's think, let's feel Agile, rather than let's do our job. In terms of things that are being missed, I would say the right perspective, the right understanding, and the right mindset.

[05:58] RESEARCHER:

Is missing most of the time is it?

[06:02] PARTICIPANT 12:

Most of the times, especially in large corporations. In startups and small companies and smaller organizations, it is better understood because people are forced to give them the shortages, either money or personnel, or space or whatever. They are forced to be more creative. They are forced to work together better or at increased levels. Whereas in larger organizations, there are strict policies in place, stricter guidelines. You can do this, but you can't do that and so forth. The smaller the institution, the organization, the higher the chances that Agile is properly understood.

[06:53] RESEARCHER:

Do you think because the culture is a better fit?

[06:58] PARTICIPANT 12

Culture is 100 percent, I wouldn't say to blame, but 100 percent because different establishments have different rules in place. And the larger the company, the more the more the organizing, the structuring, the accountabilities, the responsibilities, are broken down into departments, to individual pieces of units, to individuals as well. So that makes it more complicated, which is why you will have heard that certain methodologies have appeared, such as SAFe or scaled Agile framework. Which is in my view and not quite a few people's opinion isn't really Agile in itself because it is very complicated, very structured. Large companies prefer it because they think they are doing something. But rather than progressing, they are adding extra layers on top of each other, which makes it increasingly complicated. Whereas the whole idea of Agile is to work in simpler ways, to break down larger pieces of work into smaller ones, in small increments in order to be able to deliver value as early as possible.

[07:22] RESEARCHER:

Given that we talking about quality, I'd like you to define it. How do you define software quality in the context of agile software development?

[07:33] PARTICIPANT 12:

Yes, very important question! Software quality is first and most the product confirms to the requirements and free of defects. To achieve that we need a robust software development process that is subjected to continuous review and improvements. We don't achieve software quality without a process that implement and foster quality practices. It starts from the requirements elicitation and documentation to the testing. For example, we have review processes for the requirements, the code and the design. We have check lists and documented guidelines. It is important to also assure internal quality. Internal quality has to do with the way that the system has been built and designed. We aim for clean code. This mean maintainable code that confirms to well established guidelines. We also aim for sustainability of the design. This means the software design is able to cope with new requirements easily moving forward.

[08:32] RESEARCHER:

Ok, let's gets a little bit into details. Can you describe to us your Scrum environment? You can choose any one of your experiences or your work or project.

[08:50] PARTICIPANT 12:

Yes, I will refer to the time I spent in the online gambling industry. And the Agile setup we had there was excellent, in my opinion, because we were operating from several countries around the world. Probably nine or ten, if I remember correctly. And within those countries, we had multiple offices as well. To give an example, within the UK, we had offices in London and Leeds and also it was part of the UK technically, but we had an office, the headquarters was in the Isle of Man. In the states, which had offices in New York, California, and Texas. So, we were a very distributed business. But despite this issue, we were able to work extremely well together, all the departments, I would say. And that goes back to the thing that we covered a few moments ago, and that is culture and the mindset at the company itself. Our values, which is working together, trying to work in small increments, deliver value on a regular basis with a preference for releasing to consumers as early as possible, ideally every couple of weeks.

[10:26] PARTICIPANT 12:

But that's not to say that we didn't have significant projects that lasted a few years. But those were primarily because of the highly regulated environment we were working on. That is a real money business as opposed to a play money, like most games are nowadays, which means that we had to work with multiple regulators in different jurisdictions. And they had different requirements which required extra attention from our side, which is why I would like to point back to what I said just now. We had a number of projects that lasted a couple of years. So, yes, that's the setup that we had there. In terms of the development teams, most of the teams were using Scrum and Kanban. Other departments such as sales, marketing, or customer support, they weren't specifically using a methodology or a framework, but they were working together and discussing on a daily basis, making sure they adapt their work in progress and their plans accordingly.

[11:50] RESEARCHER:

So, let's use Agile or Scrum to refer to this setup. So, in this Scrum setup, you talked about in the development team, you were part of that, or you were somewhere else?

[12:00] PARTICIPANT 12:

Yes, I was. I was working with five different teams in Europe, a number of those in the UK and two in Bulgaria and one in Malta.

[12:14] RESEARCHER:

Can you describe to us how Scrum worked on that team? How was Scrum implemented on that team?

[12:23] PARTICIPANT 12:

We tried to cover as much of it as was initially, invented or designed by the founders, Ken Faber, and Jeff Sutherland. And on most occasions, we were successful of playing the game because Scrum as a three month, is purely a game. We were successful in playing it to the rules, but in some occasions, we had to implement specific roads, given the nature of the business. Let me give you an example because I'm thinking you would want to hear. So, the Scrum guides indicate that sprint has to end. And when the previous sprint ends, the next one immediately starts. Whereas in our set up, because we are operating, like I said, in a highly regulated environment, and we were working against extremely important deadlines such as the World Cup or the Champions League final. In some cases, we had to design our sprints so that we take into account those major events and avoid releasing around those times in order to prevent issues in production, which meant that if a particular sprint ended, let's say last Friday and this weekend, we would have a major event like the World Cup final. Then we wouldn't start the next sprint until two weeks from now. And during this time, we would be focused on immediately fixing critical issues in production, which would very likely appear given the higher number of users on our platform. And given the pressure on us to fix everything in time to make sure that we don't impact on our customers. So, for us, it was more important to take care of our customers rather than blindly starting a sprint immediately after the previous one. I hope that makes sense.

[14:37] RESEARCHER:

Yeah, it does. You talked about being successful. Can you talk to me about the factors or the variables helped to make you successful in this Scrum implementation?

[14:53] PARTICIPANT 12:

Yes, of course. We were successful primarily because we worked together extremely well. We communicated regularly and we were very transparent with each other. And that meant that our work was easily available for others to inspect. We got immediate feedback from our customers, immediate or next immediate, obviously from our customers, and all the risks were properly discovered and mitigated. That's not to say that we didn't have issues that were really unforeseen, such as a distributed Denial of Service attack, which severely disrupted our platform and put pressure on us. But that was something that teams outside of the Scrum setup were responsible for. And there was nothing for us to cover. So, we had a particular DevOps and a particular NetOps department. These departments are the ones who ensure reliability. And when this particular risk impacted, they were extremely proactive

in both mitigating and then addressing the issue, but also extremely proactive in taking steps towards preventing anything like this happening in the future. So, yes.

[16:28] RESEARCHER:

So, you mentioned that you worked together very well. What's made you work together very well. Is it because you've been working together for a long time or because you click or?

[16:43] PARTICIPANT 12:

I would say this goes back again to the company culture. One, where people are encouraged to bring in colleagues even from previous jobs or recruiting new people. And, when performing the interviews, they look at certain particular skills. So, they value soft skills. They value attitude and approach as opposed to numbers of experience. So, they want to see people who are cooperative people, who are quite [inaudible] in environments where working together is paramount. So, it goes back, I think, even at the stage of recruiting. But obviously, the culture within the company encourages people to be as open as possible with each other. And the company made it clear that no matter the consequences, obviously, except the illegal reasons, but no matter the consequences, people wouldn't be sanctioned for discussing their opinions, sharing their concerns and so forth. So, it was a very psychologically safe environment that people were encouraged and are really happy to work with their colleagues despite the distances between the offices.

[18:14] RESEARCHER:

That's very interesting concept, which you mentioned is psychologically safe environment. Can you describe to me what is a psychologically safe environment? Is it created by Scrum or it is inherent to the organization?

[18:34] PARTICIPANT 12:

Oh, that's an amazing question. I don't know really because I didn't research this. I think you can choose different topics from different explanations. So, it may be linked to Agile in itself, but it also may be linked to the company, to the founders, to the spirit they brought, and they matched it and developed in the company. I think that you've got the right idea when indicated that Agile is somehow linked to psychological safety. Because if you're not valuing people, if you're not getting your customers, if you don't want to have a decent environment where people are valued rather than politics and what not. So, yes. I think that's an excellent starting point to research further.

[19:51] RESEARCHER:

It's an interesting thing. So how does psychological safety help software quality?

[20:02] PARTICIPANT 12:

It does! What I have observed is developers and QAs for example take initiative more and invest on the success of the product including its quality. For example, developers put more effort and time on writing better code. The QAs become invested and care about the quality of the software, even the relationship between developers and QAs improves. They communicate better and share information faster to understand bugs. The team becomes more efficient at assuring quality. I saw improvement on the quality.

In my opinion, for a workplace or for a company to be psychologically safe for each employee, that would mean that everyone in the company feels like they belong there. For one thing. And they feel like no matter what the seriousness of the trouble or the impact that a particular issue may have on them or the business, they are encouraged to discuss with their colleagues, with their managers, raise their voices whenever they feel something is wrong or whenever a certain project or certain initiative is going to have a negative impact on himself or their colleagues or the company or the customers. So psychological safety, in my view, is a quite complex term that includes a wide variety of things like those I mentioned and other ones as well. And I think this psychological safety is linked to the company's success. So, the higher the psychological safety, the more successful the company and the product it builds are.

[21:34] RESEARCHER:

Fantastic. Thank you very much. We'll move to the next question. Can you take me through the journey of a user story or a feature from inception to release? How does it happen in this Scrum setup?

[21:59] PARTICIPANT 12:

In the Scrum setup, user stories or features normally come from the business representatives, the stakeholders who then discuss with the product owners or the business analyst and as a result of those, we have placeholders created on the backlogs. But one thing that many people get confused is user stories aren't the tickets that you create in backlogs. User stories are the conversations that the business wants to have with the engineers in order to produce something of value for the business. And when it comes to the workflow and once the individual user stories are created based on the interactions between the business representatives, the stakeholders, with the members of the Scrum teams, such as the business owner or the product owner or the business analyst.

[23:16] PARTICIPANT 12:

We then thought about running Refinement sessions where the engineers would look at those user stories, their requirements the business puts forward, and they would discuss those. If they had concerns or questions, the user story wouldn't be ready for development, obviously. But those that were ready for development went to an estimation session, which meant that prior to this sprint planning, we would have most user stories estimated. And, based on the decisions made by the product owners, when it comes to managing the backlog, certain user stories at the top of it were selected for development in certain sprints, which meant that they were linked to specific sprint goals. Once they went into development and went through the different stages of the SDLC, if they were successful in being produced, normally, they would get released by the product owners.

[24:40] PARTICIPANT 12:

And they would reach the end user, by being shipped to production. But obviously, the workflow doesn't end there because there may be instances where those particular user stories cause issues in production, either by themselves or in relation to others, which means that technical problems, namely bugs, are being raised against those. So that would be the end to end of our user story. A more important topic from my view is prioritizing user stories, because when working with multiple stakeholders, it is very difficult to know to prioritize and choose what you want the engineers to build. It is always a fact, maybe not always, but most of the times it's a fact that the business wants more than the engineers can

accomplish in a specific timeframe, which is why it is good to use prioritization techniques. And one thing that I recommend is for teams and companies to think about is the Pareto Principle. And this is basically something that says eighty percent of the benefits come from twenty percent of the features. So, if you think about it, only twenty percent of anything that most companies want to do is really important. Let me show you to be more precise. Have you heard of the Pareto Principle?

[26:30] RESEARCHER:

No. This is the first time.

[26:33] PARTICIPANT 12:

Pareto was an Italian economist and he discovered that eighty percent or four fifths of something gives one fifth of something else. Let's think about land. Eighty percent of the land in any given country is controlled by thirty percent of its population. It doesn't have to be precisely eighty percent or precisely twenty percent. But the idea is the same. So, when working with products, say you are a company producing soft drinks. Most of your sales would come from one particular brand rather than all of those. Think about the manufacturers of Coca-Cola. They produce quite a few drinks, but eighty percent of their revenue likely come from Coca-Cola, and only twenty percent from all the other brands of soft drinks that they have. I will look this up once we end the call because I want to verify this.

[27:58] RESEARCHER:

There are two follow up questions I noted here. The first thing is the user story in itself. You said that the user story is the conversation the business wants to have with the developer.

[28:13] PARTICIPANT 12:

Yes.

[28:15] RESEARCHER:

It can be abstract and I'm assuming, well I'm injecting my experience here, the detail can be lost in the conversations.

[28:28] PARTICIPANT 12:

Yes, that is absolutely true. Which is why if you are using a user story because user stories aren't mandatory in any given framework or methodology. Not even in Scrum, user stories aren't mandatory. User stories were basically invented as part of Extreme Programming, have you've heard of it?

[28:49] RESEARCHER:

Yes.

[28:51] PARTICIPANT 12:

So, user stories were invented by, I forgot his name, but one of the guys who put the skeleton of XP in place. If you are using user stories, you may want to use specific templates against your needs. So, let me give an example within the gambling environment I worked in. At my suggestion we introduced a standard template for the user story because we were operating on multiple markets. We had multiple customers. We had different locales, for example, in Switzerland, we had German, French and Italian within the Swiss jurisdiction, obviously. So, given the high number of variables that we had to work with, I suggested that we use a template with some things that were mandatory, such as the acceptance criteria or the license name or the locale. But for other work points, we made those optional. So, I think it's important to put things into perspective and realize that, yes, things can get lost in translation. But if you put something in writing, it doesn't necessarily mean that it's bullet-proof as well. So, you have to be flexible, you have to review your initial thoughts. You have to validate your hypothesis.

[30:28] PARTICIPANT 12:

Any given user story is just a thought, and it basically is supposed to stimulate conversations between the business and the developers. But let's not forget that during the development life cycle, you want conversations between the developers and their quality assurance analysts and the product owner or the business analyst as well. Once the engineers start coding that particular user story, if they have questions and if they want to make sure they are going the right way, all they have to do is ask for immediate feedback from the product owner or the quality analyst because in this environment, in software engineering, it is always good to operate on a test-driven approach, which is you write your tests first and then you start coding. Obviously, you want to make sure that you synchronize with your colleagues within the Scrum teams, so that the work you produce, as an engineer, is valuable. Does this make sense?

[31:50] RESEARCHER:

Yes, it does. Another follow up question is when you describe the journey of a user story, you didn't talk about quality and how do you assure quality in the journey? So my question is, how do you assure quality in this Scrum setup?

[32:03] PARTICIPANT 12:

Well, quality is something that should be thought at earlier level within the development lifecycle. But at the granular level, you want to encourage conversations as early as the backlog refinement. You remember this term I used earlier?

[32:22] RESEARCHER:

Yes.

[32:23] PARTICIPANT 12:

So basically, during the backlog refinement, you want the whole team ideally to participate so that the developers are discussing with the QA people so that they understand. They are on the same page when it comes to that particular user story. So, when it comes to validating it, they are again on the same page. So, you want the acceptance criteria to be

understood by everyone, really. So that the work produced is a value rather than waste or requires further rework. If that makes sense?

[33:00] RESEARCHER:

Yeah, it does. So, how does this engagement or collaboration helps quality?

[33:01] PARTICIPANT 12:

It improves the understanding of the requirements, what the team will develop and test. So, we see less bugs, because developers know what to develop and QAs know what to test and how to test it. So, quality doesn't happen just at the end of the work that the developers perform and once the coding is done. Quality should be embedded at every single stage as early as the user stories are created, as early as the acceptance criteria is discussed and then put on paper, or as early as if you are using a Gherkin model, or as early as the statements are written.

[33:37] RESEARCHER:

That's excellent. I picked up two follow up questions. Let's start with the first one. How do you make sure that quality is embedded in every stage of the journey of the user story?

[33:52] PARTICIPANT 12:

First of all, you must have the right mindset. You have to think proactively because there is a lot of room for error in this industry. In software, as opposed to construction engineering, there are many unknowns and you know very little when you start building it, you discover more and more stuff. Which is why you have to have certain, I would say Gates, because Gates, would suggest a very rigid, different approach. So, you must have certain, let's say, feedback opportunities so that you review quality against that. But this leads me to another topic, which I think you would like to hear. And that is what qualifies as good quality? Let me give you an example, if you have an existing product and you want to improve one of its features, then that particular improvement has to be top notch because you already have an existing product that generates money. It generates revenue. So, you want to make sure that what you add to it is of top notch quality.

[35:21] PARTICIPANT 12:

So, you don't dissatisfy your customers. Whereas in other approaches, if you are launching a new product, one that isn't available to your customers, in that case, a perfect quality is the enemy of the product itself. Because in that instance, first of all, you want to validate your initial hypothesis that the product is going to be used by your customers. It doesn't matter if it is perfect or good. All that matters is that you generate feedback from your customers. So, in this particular instance, you don't have to make your quality of the product. It doesn't have to be top notch. It can be buggy. It can be slower than you would normally have it. But you would generate feedback. And this is what startups normally do. They launch something. They say it's a beta release and they want to gather as much feedback as possible from potential customers rather than wasting time building something perfect which may not necessarily be used by potential customers.

[36:35] RESEARCHER:

Another thing that I'd like to follow up is you mentioned that QA is involved from the beginning from the backlog refinement. This is in contrast with traditional methods, which the QA waits until the end. Do you think this early engagement does empower the QA analyst? Does it change the behavior of the analyst?

[37:02] PARTICIPANT 12:

Of course. I think it empowers them. I think it makes them more aware of the importance of working together. And obviously it significantly reduces waste for the company. And let's not forget it significantly reduces the dissatisfaction of the team having to be forced to rework or redo something because the initial requirements weren't met or what they think, the company wanted to do in the first hand wasn't delivered as promised, if you will. So as opposed to the traditional way of working, the agile mindset encourages people to be proactive rather than reactive. So rather than wasting time and money for the company and for the team itself, obviously we want to make sure in an agile environment, we want to make sure that quality is built in from the first stage, which is gathering requirements, discovering what the business wants to build. Because based on that, we make plans. We make assessments. We decide how to proceed. We start drafting acceptance criteria. We start making plans. We start discovering possible ways of working. We organize around larger features, also called Epics.

[38:43] PARTICIPANT 12:

And when the development finally starts, we are in a significantly better position because we have managed to decrease the risk. And in so doing, we increase the quality of the work itself, because by that stage we know much better as opposed to having some written requirements from six months ago. And they may no longer be valid because market conditions change regularly. So, it is better to be proactive, to work in smaller increments of up to one month rather than blindly following a plan that was put forward a year ago.

[39:41] RESEARCHER:

Fantastic. Thank you very much. The next question. Do you think this was Scrum setup you've been describing to us produce quality software and how?

[39:52] PARTICIPANT 12:

Yes. Absolutely. I think it's a combination of things that we've been discussing already. The value that the business produced has been validated by our customers. To give you an idea, the business that I work for, released valuable software for the clients who loved it. But let's not forget that releasing software isn't the end of the story. So, what I'm trying to say is the company was extremely agile as a mindset because once we delivered something for our customers, we didn't claim success immediately. We wanted to make sure that the customers had the right support. So, our customer support department was very appreciated. And it was actually rated the highest and the best customer support in the industry. So, they were very eager to help our customers. The marketing as well was very eager to promote the services and the features that we were constantly releasing to production. And obviously, that meant that the whole company was acting in unison. So rather than having business units that were fighting with each other, we supported them. We had each other's back. So, let's say it's a higher perspective if you will. For the development team, Scrum reduces fractions and helps better relationships in the team. This makes the team more efficient. They communicate bugs frequently and accurately. They share their

understanding of the user stories with the QAs from the start and align the quality criteria as early as possible in the development cycle.

[41:44] RESEARCHER:

Fantastic. Let's move to some examples. Can you share with me a positive story from your experience about Scrum and producing quality software?

[42:00] PARTICIPANT 12:

I have quite a few, actually, but the one I would like to refer is, like I said in this particular environment, we were working in a highly regulated market. In one particular successful story is related to a regulatory change, which was coming from the Spanish regulator and that particular regulated wanted the changes applied as soon as possible with no impact on the customers. And when thinking about it, the business representatives, the product owner, discovered that implementing that particular requirement from the regulator would basically result at a loss for the company. But it was decided that it is well worth the effort. So, it's well worth taking a financial loss for two reasons.

[43:17] PARTICIPANT 12:

One, we would satisfy the regulators so we wouldn't risk losing the license, which is obviously very important because without a license, you can't operate. And two, you would satisfy the customers who would be in a better position once that particular change is made in the production environment. So, based on the conversations that we had, you know, as a team, as a large team within the business, we decided to basically cancel the active sprint that we had and to start working immediately on addressing that issue. That wasn't that long in terms of the work that was estimated. It was just four or five days' worth of work if I remember correctly. But the impact for the business was tremendous. Like I said, we did take a loss. The business did take a loss, a monetary loss. But the regulator was happy that we were very swift, treating their request with urgency. And also, the customers were very happy. And I can say that throughout the years we've seen the cross-sale between the different products we have of very high levels in Spain. So that was a very successful approach for the business. And it's linked to this agile mindset.

[45:07] RESEARCHER:

So, you added value, which is fantastic. Do you think it's because of Scrum or because of the performance of the team or the combination?

[45:18] PARTICIPANT 12

Oh, I'm not the expert really. I would probably say it's a combination of all these and likely more depending on the circumstances. But if you're not working against the value, against certain metrics, then you weren't looking at the right things. So, value comes in different forms and shapes. Value isn't necessarily linked to money. Let me give you another example just to make my point. A couple of years ago, in this particular environment, one of our competitors went bust. So, it got bankrupted. I don't remember the reason, but the company decided to purchase the debts of those particular shareholders, which meant that as long as the users of that particular competitor had a country for us, that we would reimburse them. The money that they had. Let me see if I can find this.

[46:45] PARTICIPANT 12:

This one. This was in 2017. So, competitor [Deleted to preserve the participant anonymity] So, the value here wasn't money. The value here was a multiple fold. One, public relations, we were showing that we care about the customers despite the fact that they were working on a competitor. Two, we showed that we are in an excellent financial state. And that shows credibility for others to keep playing with us. [Deleted to preserve the participant anonymity] en some of them probably, I don't know, ten to twenty to fifty percent, but some of them would keep playing, because they would have trust that PokerStars is the place to be. So, this was an excellent approach strategy by PokerStars. So, the value here isn't money. It is multiple fold. Does it make sense?

[48:50] RESEARCHER:

Yeah, it does. So, it is a very good example. Unfortunately, I have to move to another example. But this time is a negative story. It doesn't always go rosy. It doesn't always go shiny. So sometimes it fails, Agile. If you have a negative story would be great.

[49:11] PARTICIPANT 12:

I do have a negative story. Not from this environment because this has been great. But there one in one of my previous roles, I was disappointed to see that the information that myself and my colleagues were providing to our managers was ignored, was dismissed.

We've found too much thought, too much analysis. And basically, to put things into perspective, we were trying to alert the business that a significant risk is developing. However, they were dismissive saying that the project must continue as was originally designed. And neglecting the risk is a smart idea in their view only to turn out a couple of months later that we were right. The risk did materialize, which meant that the project had to be canceled. And that resulted in a significant loss for the company in money and time, obviously. But if we think about it more carefully. It also resulted in a loss of opportunity. So had the business pivoted, had the business realized this sooner, they would have been able to do something else. So, we an opportunity as well.

[50:58] RESEARCHER:

Yes. The last question. It is a little bit provocative question, but the purpose is to get you thinking and tell us your opinion. So, it's not to upset you. What do you think of this statement: Agile produces poor software or bad software?

[51:19] PARTICIPANT 12:

I wouldn't know really. I've never heard it discussed. It may be that some people don't really have enough feedback or haven't work in this environment, they may be tempted to say that it produces bad software. But let's not forget that, like I said, agile is a mindset. It's something that you have to think, you have to apply. OK, let me give an example, because I think you would understand better. If you just follow the rules of certain frameworks or methodologies, say, Scrum in this particular instance. If you follow the rules, you can say, yes, I'm doing something by the book. Let's say you are familiar with the sprint review.

[52:27] RESEARCHER:

No.

[52:31] PARTICIPANT 12:

So, during the sprint review...

[52:33] RESEARCHER:

Ah! The sprint review, yes of course.

[52:37] PARTICIPANT 12:

So, during the sprint review, the development team presents the work that they accomplished. And most people would say that's excellent. They're Agile, right? I've seen that on many occasions. And in my view, things could become better because what you really want to have at the sprint review throughout the event, you want to have feedback from the stakeholders based on what they see immediately. So that's what you're really after. If you are not creating after feedback, then you're not really agile because you don't want to see what others think of what you've built. If you blindly have a meeting that says, in your sprint review, you spent one hour looking good and, in the meantime, you are checking your phone and whatever. And obviously that's a waste of everyone's time. It's a waste of opportunities. It's a waste of no feedback. Why do you want to have, you want to have an engaging session where the stakeholders actively use ideally what you've done. They look at the results of the work and then they provide feedback. And that feedback is incorporated on the backlog for future development. It depends. Those who say that Agile produces bad software, maybe they fall under this category, which is they don't truly understand how things are supposed to be done and they don't have the slightest idea, or they haven't practiced in a true agile environment. That's my line of thinking.

[54:33] RESEARCHER:

Or they have experienced poor implementation of Agile.

[54:37] PARTICIPANT 12:

Either poor or completely off guard.

[54:43] RESEARCHER:

Or chaotic implementation of it.

[54:46] PARTICIPANT 12:

Precisely. You just do it because others told you to do it. I've seen some cases where project managers were forced to become Scrum masters. But they had no idea what Scrum masters do. I've seen cases where project managers became product owners, but they didn't even know the product. So how can you be a product owner if you have no idea?

[55:11] RESEARCHER:

I've seen that yet. Okay, PARTICIPANT 12, thank you very much. That was my last question. Do you have any questions for me?

[55:19] PARTICIPANT 12:

No, not really. I wish you the best of luck. This sounds like, you know, a very interesting and a very intriguing approach. I would encourage you to research further into the topic that really caught my attention. And that is the link between agility and psychological safety.

[55:47] RESEARCHER:

Yes.

[55:47] PARTICIPANT 12:

That is something very interesting.

[55:49] RESEARCHER:

Or I may study that. I may be search that by myself. Thank you very much. I enjoyed every moment of it. Thank you.

[56:00] PARTICIPANT 12:

Myself as well. Thank you so much. If you've got any other questions, please get in touch. I'm available. I wish you the best of luck with your research.

[56:15] RESEARCHER:

Thank you.

[56:15] PARTICIPANT 12:

And enjoy the weekend. Let's not forget, stay healthy.

[56:21] RESEARCHER:

Okay. Thank you.