R: The recording, and I'm also sharing my screen with you, so first of all, can you see my Screen? You should see a Google Drive file.
P: Yes, okay.
R: Perfect. So first of all, thank you for participating.
I'm conducting a study on software product quality, and specifically doing research on the ISO 25010 standard. It's a hierarchical model that says that software product quality can be assessed according to eight quality characteristics. So these quality characteristics being things such as reliability, compatibility, portability, etc. But the thing that the standard is lacking, it doesn't describe any concrete measures you can take to measure these abstract concepts. So what I did during the first part of my study, I conducted discussion groups, and during these discussion groups, I brainstormed with engineers on what data values we could find to potentially say something about these eight abstract concepts, right? So far so good?
P: Yeah, good.
R: Okay, and in the second part, and that's the part where you'll be participating, I'll enrich the information that we've gathered during the discussion group. So in the discussion groups, we found some metrics, mostly 10 to 20 per characteristic, and we're interested in finding out the source. So where can we find this data point? And we also try to find the difficulty of obtaining data. So is the data readily available? You see the dimension on the screen. So we'll assess it as low if the data is mostly available, or it can be made available within one hour or day, moderate if some additional preprocessing and cleaning is required, usually one day to a week, and if the data is mostly unavailable, and it really doesn't seem feasible to measure this value, we'll classify it as high. And then we have the technical expertise, also with categories low, moderate and high, low being just basic technical skills. So something anyone who knows how to operate a computer basically could do, or has very basic programming skills, maybe reading from a dashboard or something, moderate is most developers can implement the measurement, but technical skills are required. Because often this requires technical skills on certain tooling. And high is basically moderate, but then in-depth knowledge on a specific tool is required instead of just basic knowledge on the tool. Does that make sense?
P: Yeah, yeah. It looks good. It makes sense what we're going to do.
R: Okay, so I've gathered eight participants, so I did one quality characteristic per participant. So the one that we're going to discuss is functional suitability. Functional suitability in the standard is defined as the degree to which a product or system meets stated and implied needs. Okay? So we found a few measurements, and I'll propose the measurements to you, and you'll give me any information how we could potentially measure this, and then especially taking your context. So taking the stuff that you did at <team name>. Does it make sense? So to further specify in the context, can you first tell me a bit about what you were doing at <team name> and which tools you were working with?
P: Yes. So yeah, for <team name>, we are a platform team, so it means that we create tools and system in order to minimize the effort for the teams to use it. And then, of course, we also measure and also monitor some of our applications, yeah, and hat involves some tools that we use in order to collect data and analyze, but our focus on mainly to help other teams to speed up their process in order to develop better, faster, and with less complexity, so that's basically what we do.
R: Yeah, and as regards to the technologies that you're using, what's your tool stack?
P: Yeah, so technologies, yeah, so for programming languages, we mainly use TypeScript. There are a couple of things that we do in Python, but mainly TypeScript. We use TypeScript for the backends.

We strongly use AWS CDK, which is the cloud development kits in order to create infrastructure with AWS and also within TypeScript. And then our cloud services, our services are all hosted in AWS. So we use some of the best services, mainly serverless services like AWS Lambda, DynamoDB, just to measure some of them. But that's it about the tech stack.
R: Cool, perfect. Thank you so much. So as discussed before, we'll discuss functional suitability. Functional suitability may be a bit less technical than the other ones we've discussed. So hopefully, you have the right context and you know a bit about the measurements that we are going to do, but in case that it's not applicable for your work or if you don't have any information, we can also say that either the source, the difficulty of obtaining data or the technical expertise is just not available or not within our scope. Okay?
P: Okay, cool.
R: So this is how we're going to work. I'll present the measure, I'll describe you what exactly the focus group meant with the measure and then we'll describe the source, the difficulty of obtaining data and the technical expertise required. In one of the focus groups, someone said you could measure functional suitability by counting the number of changes you need to apply to your product to fit to original requirements. So a number of changes that will output a numeric value. So the thing we're interested in is finding out what source, how do we gather this data
P:Would that be like, for example, the impact of a specific application? Like, yeah, this is what I'm pushing or changing. Is it like the number of change? Okay, so then, yeah, that's good.
R: So where could we find this information? The number of changes needed to fit to original requirements. So what information, what's the information you need? What do we need?
P: Yeah, okay.Is there something that we talk about specific tools because I know a tool that can do it for us?
R: Yeah, exactly. We can talk about specific tools and then in my research things, I can further make an abstraction if the tool fits to a certain category of tools. For instance, JIRA fits to project management tools, et cetera.
P: Yeah, I see. Okay, so there is one thing that we kind of use in this situation, which is the AWS CloudFormation changeset. You kind of know what I'm talking about.
So that's, I'm not sure if it's the exact same thing, but Cloud Formation changeset help us to identify, yeah, this is what I'm actually developing and pushing and changing. And these are the list of the changes.And what would be the impact, for example?
Yeah, I'll be removing this or adding this or renaming this. In that way, we can easily see the impact of everything that I'm changing, like all the list of changes. And then we can evaluate if that's really something that we want to be deployed. And then we can enumerate this in risks, for example. So you can do a lot of things with that data.
R: Okay. How would you know, then, if it fits the original requirements? Where are you getting your original requirements from?
P: Yeah, so that's not linked to that. It's something that, yeah, you can easily see the technical changes, but no, you cannot compare with the business requirement, for example, the original requirement. Yeah. It would also be interesting in seeing if a change fits the business requirements.
R: What information would we need for that as well?
P: Yeah, so. More of the technical aspects of this?
R: Yeah, I would say so.
P: I'd say so this is more in the technical side of things, like everything that is being changed

in that application. But the original requirements, well, that lives, in my perspective, in my experience, that lives in the project management tools, right? As you mentioned, like Jira or GitHub projects, those kinds of things. That's how developer lives usually is. We get the requirements, and then we look at the project management tool, and then we start applying to the technical aspect of it. But I would enumerate Jira, for example, or even GitHub. GitHub can also help us in that sense.
R: Okay, cool. So to reiterate, to really measure this, we could use a combination of looking at the project management tool, looking at the features, and then for the implementation, we look at CloudFormation templates, and we can see whether the change actually fits the requirements.
P: Perfect, yes.
R: Okay, so we've now found the source. Where do we get the information from? How difficult is it to obtain the information from the source? So is the data already available in these sources? Do we require pre-processing? How much effort would be required to do this?
P: Yeah, so for the CloudFormation side, so the technical aspect of it, I would classify as moderate because, well, it's not low because, yeah, you kind of need to understand how the CloudFormation service works just a little bit, but it's not high because it's not very complex to measure this. So it is moderate because you need to understand how the tool works, but the information is easily accessible while you can see in the console.
Of course, you need to log in and then go to the console, so it needs knowledge on how to reach that data, but the data is there.
R: Okay, cool. That makes sense. Thank you for the elaboration as well. And then we have the technical expertise required. So to set up the measurement, to find out, to extract this number from the systems, how much technical expertise is required?
P: Yeah, to extract the meaning for consuming that data, for example.
R: Exactly, yeah, consuming the data.
P: Yeah, so, yeah, this is not very trivial, I would say, because, well, you want to extract, if you want to, for example, have that data available to consume somewhere outside the console, you might need to write a script for it, for example. So I would say it's not very trivial to get that extracted. But it's also moderate to be something, I don't know, I would say, yeah, high, because it really demands advanced technical skills because you need to write a script for it because there is no, like, a magic button where you can use.
But, yeah, so you actually have to have in-depth knowledge on how cloud information templates work. And also, if you're using a project management tool, how to consume maybe an API from GitHub.
R: Yeah, exactly. GitHub or Jira, right?
P: So if you want to, I'm not sure if Jira has something easy that you can extract, otherwise you need to write something that communicates within the API. So, yeah, it's the kind of advanced skills that you need, yeah.
R: OK, cool. That makes perfect sense. I'd be ready to move on to the next one.
For now, is it clear what tasks we're doing, how the format is? We'll do this 14 more times.
P: Yeah, yeah, that is good. Well, the first time now I kind of understand better.
R: Yes, perfect. All right. So next up is the market share. So basically someone proposed that your product is more functional, suitable. If it covers more of the market, if you have more of the market.
P: I see.
R: Is this something that's even applicable within our domain?

P: You mean, for example, when you can compare to what the market is doing or something? Exactly, you compare it with competitors.
R: So, for instance, looking at cloud platform usage, I think that's a great example. AWS might have a market share of 35 percent. Azure might have a market share of 32 percent. And then GCP has a market share of 30 percent. And then the rest is just smaller clouds. Yeah. So how much of the market do you take up as opposed to your competitors? Is this even relevant for our domain?
P: I would say so. Well, for example, for our domain, I would say this is not very relevant because we cannot control that. Right. So let's imagine that we are using, for example, AWS. But I don't know, there's no reality. But let's say that AWS is not the most cost service to use in the market. Yeah, that is that's not something that we usually are aware of it or do something about it. So, yeah, but for the it's specifically for the products that you're building, not the tooling. So, yeah, I would say so market share in one of the products you're you're building. Yeah, I would say so. Also, well, again, it depends on what you're building. So if you're building, for example, a platform, as is my experience, a habitat,we are building a platform, a data platform. Yes, we need to do researches and compare with the with the market and like competitors and also to understand and learn from them. But yes, it really depends on the situation. We can vary from situation to another.So, yeah. Yeah.
R: OK, so what would you say then for the source?
P: Yeah, so if it's specifically talk about research, I think the source would be pretty much the internet, I would say, because, yeah, I mean, if you want to go, there is no like a magical place where you can say, OK, this is this is how the market share for cloud services, for example, or data platform. So we need to actually collect metrics from different sources. So, yeah, it's like researching, asking questions, for example, in the community or GitHub projects and everything. OK, so as the source is very situation dependent, can you say something about the difficulty of obtaining data and the technical expertise, or is that also dependent on the sources? I'm not sure, though, because if you compare the product in the product level, yeah, it really depends.But a technical side of things is more straightforward, I would say. So if you look at sources, for example, I would say, I don't know, GitHub issues, GitHub forums, those kind of things, it's quite easy to find those kind of information, also like NPM packages issues. So, yeah, then we can go all the way to a lot of sources.
Yeah, so if I understand you correctly for the technical expertise required, it's pretty straightforward. But for the data, it really depends on what sources you need, right?
R: Yeah, exactly.
P: But, well, if you want to find a middle term, then I would say moderate, right? Because, yeah, you kind of need to understand what you're looking for, of course, so you need to understand why you're comparing from. Yeah, so, yeah, it's moderated, yeah.
R: OK, and then for the technical expertise?
P: Technical expertise? Yeah, low, I guess, because, well, there's nothing they need to know very complex to do some kind of researches, right, so.
R: Yeah, that does make sense. All right, let's move on to the to the next one. We've got 15 in total and we need to fit them within an hour. So, yeah, I think I think we can do it. Yeah, OK, so the next one is the number of unsolvable issues sent to customer support. So here, the specific thing is that someone proposed that if you send an issue to customer support, that is OK, because it might potentially be fixed. But if it cannot be fixed, that means that your product is not suitable for its purpose. Okay, so here, the measure that we count is the number of unsolvable issues sent to customer support. And then customer support can be any team developing the application or maintaining the application.

P: Yeah, yeah, unsolved tickets. Well, I would say, again, project management tools, right, because, yeah, if we have such an environment where we have, for example, a product owner, a product manager or something like that, that does the bridge between dev and business. Yeah, they started to create issues. So, yeah, project management tool is the place for us to evaluate this.
R: OK, cool. And in this project management tool, how difficult would it be to obtain the data? We specifically need to know that the issue is unsolvable. How would we know if the issue is unsolvable as well?
P: Yeah, so we kind of need to compare, like, for example, the time when the issue was created and how much time it stayed in progress, for example, or unresolved. So we kind of need to understand the status of the ticket. So not very difficult. So everyone can do it without any specific knowledge of that tool. Well, it depends on tools again, but, that can easily be accessible.
R: Yeah. So would we then classify it as low, available within one hour or a day or moderate?
P: Low, I would say low, yeah.
R: OK, cool. And the technical expertise for setting up this measurement, for really consuming this data value?
P: Yeah, it's consuming the data. Yeah, I think I would again moderate because, yeah, you need to or the other one to classify as high when you communicate to APIs. So I would say the same. I would say high as well, because if you want to consume such a data, you need to consume via script or API. So it demands skills.
R: Yeah, OK. And then the number of issues, which is basically the same thing, but then not unsolved well.
P: Yeah, it is the same.
R: Copy paste it, right?
P: Yeah, copy paste is pretty much the same.
R: One thing that we've talked about for the technical expertise required. Right now, we are always assuming that we need an automated way of obtaining the data. What if there's just someone reading the data? Is that then also high technical expertise?
P: Reading you mean consuming? You do not need an automated way of consuming it. OK, yeah, if you don't need to do something very fancy to consume it, someone reading yet, then then then it's low because maybe it would be good to have to distinguish between this. True, low manually and high automated maybe.
R: Yeah, automated, yeah.
P: Then that looks so better, yeah.
R: Yeah, to add a new level of granularity, OK, so the next measure is. The number of functionalities you've delivered divided by the number of functionalities required. So let's say you if you need to build five features, you make sure your application that someone can log in so we can change your password and more functionalities. And you've only provided these two. That means to score two out of five. OK, yeah, so where do we get this number? The number of delivered functions and the number of required functions?
P: Yeah, so I don't want to go again, but I would say project management.
R: Yeah, no, I think that's a very good thing and I think that's one thing that we've had for the interviews so far is that most often the multiple measures within the same quality characteristic have the same source. So for instance, when measuring reliability, people often said, oh, you need to do this in the code base or you need to implement logging at this place.

So the source would often be the code base, whereas this is a completely different characteristic.
P: Yeah, it makes sense that it makes sense that different metrics within the same characteristic have the same source. Yeah, exactly, because when you say unsolved tickets and resolve tickets and deliver functions, I'm thinking about tickets only. So function, for example, functionalities are also a ticket, right? Because it is it was lived somewhere in the project management tool is the same like as feature could be unsolved ticket as well. So it's pretty much the same.
R: OK, cool. And then how difficult would it be to obtain this?
P: Oh, yeah, I think it's pretty much the same of the other or the same in the first one where we need to compare. So we need to compare how much was created and how much was delivered basically. So like the status creating the status done, for example.
R: Yeah, so basically the same thing.
P: Yeah, basically, yeah.
R: OK, next up, we have quite a similar one, but. There is one thing that's really important to understand here is delivered functions divided by the functions that contribute to main goal. So the participants here stated that apart a product should only have one main goal. So let's say in the session they said, let's say you're building, you're building a a website in which you should be able to order yellow balloons. So all the functions that you've delivered should contribute to that main goal, buying yellow balloons. That's what they proposed. Yeah. So maybe it's a bit different to measure. What would you say about this?
P: Yeah, well, yeah, I would say different. I'll say a bit tricky to measure because, yeah, I mean, for example, for product management tools, you usually specify a goal and then you start creating tickets. But I don't know exactly a way for you to link. OK, this is not the features that I'm delivering that contributes to the goal.
So how many features you deliver that contribute to the goal that you set is not very trivial to get. But the source, again, project management tools, because where this is where it lives. Yeah, maybe as you've indicated, not available is something as well. If you say, yeah, no way to track whether a function contributes to the main goal, that means that it's not fully available. yeah, because, again, it really it can depends, for example, maybe Jira has one way GitHub doesn't. So it really depends off the tool. Yeah. So, yeah, it makes sense to put not available because it might be not available.
R: Yeah. OK. And then the same two questions.
P: Yeah, difficulty, I would say it's medium because it's not very trivial to get the data from. Yeah. And the technical aspect is required low because you only well, we need to understand how to click and then how to navigate to the tool. But I would say low for mainly. Yeah. Yeah. And again, yeah. High for automated.
R: OK, cool. And then we have the number of clicks before an action is finished. So someone someone proposed here if you need to have a lot of clicks to achieve what you want to do within the application, the application is probably not that suitable.
So how would we track this number of clicks? How would we get this get this value?
P: Yeah, so there is a tool called Google Analytics, which is like the most common and the most used tool to measure that it basically creates a graph from everything that you navigated through the system. So, yeah, you can easily find that from analytics tools such as Google or AWS.
R: So, yeah. OK, so analytics tools. Yes, makes sense. And how difficult would it be to obtain this data from the tool and how much technical expertise would be would be required to set this up?

P: Yeah, I would say low and medium, because we are difficult to find the data. There's no difficult at all because medium you need to understand how the analytics tool works.
R: Yeah. OK, that makes sense. I think we're doing very well. Is everything OK so far?
P: Yeah, it has been good.
R: Perfect. So someone proposed as well that you're looking at the percentage of succeeding tests. So if more tests are succeeding. Your product is probably more functionally stable. Someone said that makes sense.
P: Yeah. But also depends on the kind what kind of test, because there's tons of kind of tests.
R: Yeah, but here they just take the highest level of extraction. So you count all tests and you count how many have succeeded.
P: All right. Yeah, I would say then the CI CD tools, like, for example, as GitHub, for example, is one way, right? Because then you can you can publish somewhere how many tests you ran and how many tests succeeded. So yes, CI CD tool is the best way, because before deploy, you run the test and then you can easily see it. Yeah, you can you can extract it from from there.
R: Yeah, exactly.
P: Also, very easy to find the data. So no technical expertise. I would say medium, because you actually need knowledge on CI CD test, for example. But yeah, you really need to know where to dive in to find this value.
R: Exactly. Yeah, OK, I think we're doing very, very well. Let's let's move on to test coverage. So that's basically the percentage of your code base covered by tests.
P: Yes, this is yeah, this is I would say this is a unit test then because they need to see how many how many codes are covered by test.
R: Exactly.
P: Yeah, it's a part of the previous one. Yeah, so the search, there are tools as well to do that search. I would say there's Sonar, for example, Sonar Cloud, where we can get the data. So like a code, how to say, called reliability to something like that. Yeah, Sonar Cloud is one difficult. That's a low because the data is easily accessible there. This is what they do, basically. Yeah, and then technical expertise is also in between medium because you need to navigate through Sonar and understand what you're looking for. It's not just yeah, it's not a very intuitive UI. No, it's not intuitive and you do require some technical skills to navigate through Sonar Cloud. But as it's such a critical feature of Sonar Cloud to provide this, you do not really require any in-depth in-depth knowledge on Sonar Cloud, you just need to know how it works.
R: Exactly, yeah. So the next one is interesting. As someone said, you can take a screenshot. I think it's very innovative. You can take a screenshot, compare it against the desired UI design and then compare the pixels. So how many pixels? OK, and then if more pixels, it's functionally better suitable, someone said. OK, that's interesting. That's really new to me. I've never seen that.
R: Yeah, but if you have no clue how to do this, if it's new for you, that's OK as well. Then we'll just I have no clue.
P: I had to be honest. I've never seen something like that. Yeah, so let's let's just put not applicable. So we don't make any just guesses.
R: But I thought this was a very interesting one as well. Yeah, OK, so next up, someone proposed just the uptime. If your application has a higher percentage uptime, it's probably more functionally suitable.

P: True, that could be. So CloudWatch, the metrics tool from AWS is a way of getting metrics such as availability or success rate, those kind of things. Yeah, so CloudWatch is the perfect place to get that.
R: OK, and how difficult would it be to obtain this and how much technical expertise would you require to do this?
P: Not very easy because there's no uptime metric, for example. So you need to compare. You need to do a comparison between success and everything. So medium, I would say, because you need to actually to get that to search for that. Technical expertise is also medium because you need to navigate through CloudWatch and understand why you need to compare. That does make sense.
R: All right, next up, we have another one related to the issue sent to customer support. But here we also need to count the number of users of the application. And the reason that someone proposed this is because they said if every user of your application is complaining about the application, it probably doesn't serve its purpose too well.
P: OK, yeah.
R: So we then again need the issue sent to customer support while we've established that these can be finding project management tools and they're low difficulty of obtaining a low technical expertise if doing it manually. But now we also require to know the number of users of your application.
P: Yeah, number of users, I would say that's also from the database of an application, right? So it could be like the user database, for example. Project management tool for the issues and database application database for the users. Unless you put some somewhere, but the source of the search would be the Dynamo or the database.
R: Yeah, and how difficult would it be to obtain this? And then again, what technical expertise is required?
P: Yeah, I would say no, for how easy it is, it's very easy because you can easily find there. Technical expertise, I would say medium, because if you're looking for database, you might need to run some queries, for example, in the database that demands SQL or no SQL knowledge.
R: Cool, let's move on to the next one. So in the next one, someone said.
You can fit your implementation into a large language model and then you just prompt it with does this fit the goal? So you give the give your code bases input, you give your maybe your tickets or your acceptance criteria as input, and then you ask the large language model, does it fit the implementation?
Does it fit the goal?
P: Interesting, yeah, that's interesting because yeah, because that's that that can be a lot complex because you need to train your or your large language model. You want to understand that based on your inputs or you can just go to charge your PT, for example. But if we want to be more generic, yes, first, I would say, for example, chat GPT, they won't have the context of the history of the application, for example. Yeah, that could be one difficult to obtain the data low technical expertise required also know because you just need to literally type. You just need to copy paste it into an LLM and you you'll get some sort of response. Yeah, so we're talking about like asking LLM or not talk about training LLM that we're talking about asking an existing LLM, so not training your own.
R: Yeah, perfect. OK, so next up, someone said you can look at if a sprint review was used. So if if a sprint review was used, you probably validate your work a bit more and that makes your and that makes your your product more suitable. So how do we figure out whether a sprint review was used?

Is that locked somewhere or?
P: Yeah, it's that's kind of tricky because, well, product manager too doesn't doesn't save this information about ceremonies, for example. Yeah, so if that was used, it means like the meeting happens, for example. Exactly, but I don't know. I mean, I would say then outlook, for example, is one that we can check that, right? Because then we can easily see if the meeting happened or if the meeting existed. Yeah, and easy to find and demands no technical expertise. So low, low.
R: OK, and then we've already gone to the final one. You can see usability, see usability under here. And that's what <name colleague> will be doing with me in 25 minutes.
So for us, the final one is a boolean that indicates whether your implementation fits acceptance criteria. And this might be an interesting one.
P: Yeah, that's interesting one. I would say this is also part of the product management, too, because when you create a ticket or functionality,
you usually specify the acceptance criteria.
And then in order to put something to done or defined to finish, then you need to match the acceptance criteria. So you can easily find all the tickets that are done.
It means that it matched the acceptance criteria already.
Yeah. So project management, too, is that is the place where we can see this.
Yeah. OK. And how difficult would it be to obtain this and how much technical expertise would be required? Yeah, low and low. It's really easy to find it. Then it demands no technical knowledge.
R: OK, cool. Well, those were all that we're discussing. So thank you so much. I'll stop the recording.