R: Okay, so I have started the recording. First of all thank you all for joining me in this session. The session will be 1.5 hours and it will be recorded. Do you consent with the session being recorded?
All: Yes
R: Alright, thank you. The data will be handled appropriately as also stated in the informed consent form. So the schedule for today: first of all, I will start with a brief introduction to the study. What we have already done, and then mainly we will be focussing on two tasks, which I will elaborate on later. So the goal of the study is to learn more about software product quality. There are many models that describe guidelines for building good software, and one of them is the ISO 25010. How does ISO 25010 work? It has 8 quality characteristics. Such as functional suitability, performance efficiency. We will dive into those later. The goal of the study is to find objective measures for measuring these quality characteristics. A current shortcoming of the standard is that there are no measures, and the measures that are found in previous studies are subjective. What is subjectivity? To briefly elaborate and give an example. For instance, when looking at reliability, a subjective standard could be to ask a user of the product how reliable do you think that the software system is? While an objective standard is data that can be actually measured, for instance the mean time between failures.
P1: The reliability of the system, and not of the code?
R: Yes exactly {P1}, so we do not focus on software quality, but on product quality, so the quality of the product and not of the code. While that can contain the quality of the code, but we are looking to evaluate the product as a whole. Okay? To further elaborate on the ISO 25010 standard, eight quality characteristics, and each quality characteristic is divided into sub-characteristics. During the first task we will try to discuss and list as many objective measures as we can for the quality characteristics. And afterwards, in the second task we will try to link the ideas that we found to the sub characteristics, because the research is also interested in finding out whether the sub-characteristics are appropriate, so that is why we are not introducing them at first, we are introducing them later. So that is what we will be doing during the second task. And during the second task, if we due to the introduction of sub-characteristics, we find new quality measures, we will also add them to the notes. Are the research goals in the tasks to be performed clear for you?
P1: You do assume we do not know all eight characteristics right?
R: Yes, that is why you can all come up to this board, and for each characteristic I have written down what the characteristic is, and what it means. So we can also have a discussion on that. So for now, for the first task, you could all come up the board. I have provided post its so we can have a discussion on how exactly we could measure these things? It is important that because it is a brainstorm session we just write down as many things as we can, and later we can remove some if they are not eligible. And it is also the goal to have a discussion on this. So we are starting off with the quality characteristic functional suitability. And in the standard, I have used the standard definition in the official standard, functional suitability is the degree to which a product or system provides functions that meet stated and implied needs. Okay?
P1: So how it fits with what the customer wants?
R: Yes, exactly, that is a good understanding
P2: So how to measure that
R: Yes, that's a complex thing. That's what we are looking for
P3: One thing that pops up into my mind: if you are not happy with the product you are likely to want to change the product
R: Yeah, right, okay

P3: So if you have a finished product and want to measure the fit that could be the number of changes that need enhancement, not necessarily, but changes
P2: Large changes I guess, like batches. But really functional changes
R: Okay
P3: Typically, if you have not the correct captured functional requirements beforehand, is that a product owner or user states what he actually wants after the product has been finished
P1: I would say it is almost the opposite, like, the more big changes you have the better you fit what the customer wants
P3: At the end, yeah okay. But I would argue that if you are not happy with what you have implemented correctly, to fit the requirements, but someone changes it
P2: This is the static quality of the product. Is there a point in time where you can say, it is fitting the product right now? Or is it a continuous thing?
P1: I think it could be both. Google has looked the same for over 20 years because it works. So if you have a good product, maybe changes on that level won't happen often. But yeah, it depends on what you define as a major change
R: So you would say that then the measure would be the number of major changes required
P3: Changes to the original requirement
R: I think that is a very good point, then you really have a point of time of comparing it. Should I write it down? Number of major, and I will underline major changes needed to fit original requirements.
P1: Yes.
R: Okay, so that is one way of measuring functional suitability. Would we have more?
P1: I think market penetration could measure it. So not sure if that is actually the correct term, but especially in like an internal situation, you should know how many users you could have, and how many users could use it. So if you have like 50% of people that could use it are using it, it is less suitable than lets say 80%. And also to get to the Google example, both of those you can measure. And it doesnt only depend on the quality of your project because it is also marketing and that sort of stuff, but it could be a measure of like a quality
P2: The combination with other measures could be useful
R: So how could we summarise this finding?
P2: Market share of certain products? So market penetration may be a better term
P1: I think that means what I think that means but I am not exactly sure.
P3: The customer satisfaction: the amount of people using it / the amount of people that could be using it, is a very rough ratio
P1: *Inaudible*, type of product you have, and the type of product you have. For some products it can be infinite, but like for health care providers in the Netherlands, you have a very hard number
All: Yes
P2: A kind of measure for customer satisfaction. Like an NPS score maybe
R: That is called an NPS score?
P1: Net promotor score
R: Right
P2: I just learned the term
P1: It is basically how likely you would be to recommend it
P2: Net promotor yes
P3: If something is not functional, maybe you also have a lot of service requests / help requests
R: Okay

P3: You can argue that if a product is not functional, then you
P2: You can define a ratio as well. Suppose you have an X amount of users and an Y amount of requests, if it is one it is not good. If all your users put a help request something is wrong. A very rough measure.
P1: I would say you mainly relate it to UX issues.
P3: If the product is not a functional fit you can't or you don't know, or it is not clear, how the product can help you, or does not fulfill a functional need, or it is not clear, you could put a service request maybe. In some form.
R: Could you then add the dimension, you are mentioning a very valid point but it could also be UI related. So you could say number of service requests related to product functionality maybe?
P1: Yes. But how can you define functionality?
P2: For example, if you do not have a way to reset your password yourself, that is a functional emission
P1: I would say thats mostly UX
P2: No, it is a functional emission in the product. You have to access the service desk on a simple thing like a password reset. I think it could be a measure
P1: I am not talking about UI right, I am talking about UX. It is part of UX
P2: It is a functional emission
P1: But if you say functional emission, I think it is very hard to quantify
P2: It is on whether the system provides functions
R: That meet stated and implied needs
P1: But you can say, hey lets measure how many functional emissions we have, but basically. Yes of course, it is a really easy way to get to the point, but it is also very hard to quantify.
R: Should we just add it?
P1: User input on like
P2: If you do not have a functional request, then probably the product does what it does well. So yes.
R: So how could we summarise this?
P1: I would say amount of unsolvable issues going to customer support. Because they are not solvable because functionality is missing. But I think actually quantifying it as unsolvable would. Like a password you said is not unsolvable, because there could be functionality
P2: But then you miss it, that is the point. The quality of a product, as an example
R: Okay, so looking at the degree to which a product or system provides functions that meet stated and implied needs. Could there be more ways to measure this? No is also an answer
P1: I can not think of any
P3: I think objective is it hard to think of measures, but we have a few good ones to quantify, with ratios and numbers, which when added together might be useful.
R: I think the protocol for this session is that if we think we have a complete list of measures, or we cant think of more, we move on to the next one.
P2: Okay but, It sounds like you have, old fashioned, a design or list of requirements original, and you have a product that is finished with a number of functions, and you would have to match the functions in the product, how do they match the functional requirements. You could measure all the functions in the product against the original requirements. So number of functions provided, implemented versus the number of functions envisioned or required. So you have a list of requirements.
P1: But that assumes some real protocol development

P2: Yes, I said old fashioned. You have implemented 80 of the 100 functions, so you miss 20
P3: But then if you do 20 along the way and use user input, it is hard to quantify what came before and what came after
P1: But if someone came to me with a measure like that, I would say yeah it is useless. I would say then define requirements clearly at the start, it has more to do with the requirements than the actual product. To measure the initial requirement quality, which is often worse than the product quality
P2: So lets say you have a list of requirements for the user, and you give it to a supplier, and they start building it, and they give it back. And you can say, nice, but two out of ten functions you have not implemented: password reset. You have not implemented a dashboard for example. I miss it. You can measure that. It could be an okay measure
P1: Sure, but it is not how I would like it to be measured. But we can add it, it is a way to measure it
R: It could be, but we are questioning it. So maybe, because we have lots of discussion on this one, I should add a note here that we are unsure whether on this measure. Maybe that is the way to go forward. Okay, everyone okay to move on to the next quality characteristic? So next up, we have performance efficiency, which is the performance relative to the amount of resources used under certain conditions.
P2: Of the end product?
R: Of the end product yes
P2: Interesting yes
R: The product of the product also considering resources. So a product that does well in this category performs well, so has a high throughput while not using all of your hardware resources for instance
P1: I would say CPU Utilization of your hardware, something like that. Or like, utilisation of your hardware
P2: All the metrics that relate to hardware
P1: Like is your memory never going above 20%, then you are probably not being efficient there.
P3: Also hard to quantify, because it is also about scalability. If you have a double load, or double input, it is not more than linear lets say. If you have double input, your processing is also not doubled. So it is not the square of.
P2: So thats for the functions right, for algorithms it is quite easy how they perform. But for a whole system. So basically the scalability. I think I agree, but how would you measure that?
P1: There can be more than one right.
R: Yes you definitely can. So I have written down hardware utilisation, so we can divide those into several sub-characteristics even, such as memory usage, cpu usage, network usage. But we can have multiple, so maybe we can elaborate a bit more on scalability. How would you measure this because scalability is still very latent
P2: Like when you double the load suddenly, discrete doubling of the load. How does the system handle that?
R: Right
P1: What is the .. . I would say maybe put it in latency until scale is achieved. So I am spinning up a new VM for this user, how long does it take like if someone from Russia accesses my website, for cache to build up. What is the latency until scale is achieved basically? Make it very objective, there are more for scalability
P3: I think one of the characteristics could be the response time. If you have a good product, then the response times do not change when the load is increased. So if you have

ten users or a million, you still have a good response time, then you have a scalable product from the end users perspective
P1: But how to put in in a measure
P2: Suppose you go from one to two million people using it, if the 2 millionth person has zero latency compared to the first one, then it is perfectly scalable in a way right. There is zero downtime for spinning up any new
P1: You can make it very mathematical. What is the nth user load time compared to the nth plus something user and just
P3: So the cost per user should not increase. If one user costs one ten cent, if you have ten users it will be ten million cents at the maximum
P1: I think you want this to decrease
P2: You can define a scale in fact right
P1: I think these are two separate measures
P2: Okay, so latency per user is a good one I think. Latency per user independent of number of user
P3: How does the latency change with decreasing load?
P1: It needs to be independent or decreasing. For a website it could make a lot more sent the more users the better it works. But it can't be worse, because then you are not scalable. That is the measure. The same for costs, it needs to be stable or better.
R: So how would you put this into words?
P1: Let me.. So the.. The cost is easier to produce. The cost per user does not increase with the amount of users. You can measure how much it increases or decreases and that is an actual message, cost per user, you can make some acceleration almost.
P2: Exactly, I was thinking that is a second directive with timers
P1: And you can do the same with user latency as well
P2: So a function of time and load basically
P1: And I think one of the main things you will run into with this is that it will scale pretty linearly, until it hits a bump, because something in your infrastructure decides that a million users is not a great idea
P2: There is a very discrete step for erroring probably
P1: I think that is important as well, having your maximum amount of users be related to the maximum amount of users you would expect as well. Can you handle your maximum expected amount without an exponential increase in latency?
P2: I think that is a good one as well, so the measure of maximum you would expect times two or three or something. Can you still handle that or not?
R: Okay I have put these down. So we have gone into hardware utilisation, we have gone into the latency of users should not increase, the same for the costs. It can be plotted as a function
P2: Performance relative to the amount of resources used
P1: I would say that this could also, no it would fit better under another category that could come up. With how global it is.
R: What do you mean with how global it is?
P1: Is the experience the same, is the performance the same for someone here and for someone in the US basically
R: Right
P1: Is your caching global
P2: Is it location independent

P1: Is it location independent, but can you measure this? Because it won't be. I think there will be another category where it fits better
R: It would be good to write it down, later we could always rearrange, so how would you write it down? Maybe it would be a good idea if you write it down so you get it in your own words
P1: Maybe P2 writes it down so it is readable later
P2: I will try. Independence of performance
P3: The measure is the performance per location
P1: It is something you can actually measure. What is the ping from. It is actually quite complicated to measure.
P3: You can maybe, because it is relevant to ask because we only have an app. We do not really have it
P1: We have an appsite
P3: But if you have something which is global, you could not use this measure
R: Are we ready to have on to compatibility? Compatibility being the degree to which a product or system can exchange information with other products or systems.
P2: Basically quantify it for example your API performance. Or is it not about the amount of information? Just if it can be.
P1: Systems are not users in this case right?
R: No, systems are not users indeed
P3: It is a kind of, how do you measure the adherence to standard, which pops up into my mind. If you have a product that you can integrate it. The user adherence to standards
P1: I would start with do you have documentation for your API's. Yes or no. There is very standardised API documentation. Do you have standardised API docs?
R: Yes, objective measures can be boolean factors
P2: Another thing is if all your interfaces adhere to.. The number of interfaces, because it should be a measure, that adhere to standards
P1: Yes
P3: So we do what the *organization part* controls right. Controls for certain frameworks and how much of these controls have you implemented.
P1: But this can be just yes right, I have an API doc
P2: You can make it proprietary. You can make a bespoke API for authentication for example. You can also decide to adhere to standards, to OAuth. But you can also think, oh I make my own authorization standard and document and say, good luck
P1: Okay, sure that could be a good one. Do you use one of the standard authentication?
P2: Do you have standardised the interfaces that adhere to international or industry standards?
P1: Hmm, I think we need separate measures for these types of things. I think a lot of authentication could be separate, but I think you have like a lot of things, a lot of things have some sort of API behind them. I think this is for inter platform, or inter product API's. I would say how many of your features that are reachable for an end user are also reachable by API. These are two things you can divide and make and actual measure of. How many things that are in your API are documented in a standardised way? It also gives some sort of value
P2: You can argue if compatible products, because you can also access it by API.
P1: Yes, because that is what I am looking for. If we are having a lot of issues with products, because the API docs are in some sort of, I really forgot the name, API standard
P2: Swagger
P1: I think something like that is super important for this application

P3: Then you can argue, the number of features exposed via API, because then you make your product usable or compatible with other systems. Otherwise you only have a user interface which is a human computer interface, then you have a system system interface, because you have features exposed by API's. And you have to document those API's using standards like Swagger.
P2: So that would be a ratio as well. You can divide it
P1: It results into ratios. Are you writing it down?
P2: So the amount of API functions divided by the amount of actual documented functions. What was the other one?
P1: I think we have the amount of user available functions divided by, or the amount of API available functions divided by the amount of user available functions, that value can be bigger than one because you can have more API functions, which is probably great for this category but not for this category.
P2: So for users and for other API's
P1: And I think the authentication point. Do you adhere to standardised authentication protocols? Because I think that is that all applications on the internet are integrated with some sort of authentication level, or should be. This is always a mess, so.
P2: Did you already install the new company app on your smart phone? They told me to just experience it. So authentication, and then the quantifiable measure is
P1: Is authentication standardised, yes or no? Like you have Oauth, but I do not want to specifically write Oauth, because there is more than that. If you are using some sort of **inaudible** or Oauth, or whatever you are using.
P3: Please adhere to certain standards, do not do it yourself
P1: But this makes the measure difficult right, because what is standard? Because you can just invent a new standard
P2: You can find it yourself, yes
P1: There are several things we promise we do not do ourselves, yes.
P3: It can also be if you exchange data with the API you also adhere to industry standard data formats, like EDI. So don't create your own formats for exchanging data.
R: Yes
P2: There are some standards for this basically, like comply with industry standards
P3: Comply with industry standards data formats
P2: The answer would also be yes or no
P3: Or the number of interfaces exposed divided with the number
P1: You could just say amount of data sent divided by amount of data sent that is conform industry standards
R: I think that is what you can do with all these boolean things
P3: Industry standard data
P1: I think one of the problems is also, maybe this will come up later. How compliant is your data? Are you sending information you shouldn't sent to other programs? But this will probably come up later.
P2: Will encryption also come up here? Hmm no it is more of a security thing or usability
P1: Well encryption is also compatibility, because if you are not using AES, a lot of things will say
R: Okay, so moving on to usability, which is the degree to which a product or system can be used by the user to achieve goals with effectiveness, efficiency, and satisfaction.
P3: This might be the most subjective one out of them

R: That is why we are specifically focussed on objective measures. In the industry there is already a standard together with the standard that provides measures. But especially for usability they are very subjective. That is what this research tries to do.
P3: For usability, it is more like, there are some standards for, but it is more like does it work for disabled people, or for hearing, or visual issues. How do you call this?
P2: You can sort of make a measure for other products or programs if they have the same functionality and see how people perform a task for the product. How effective is that? How does it compare to our software, some sort of like a control group? I suppose scientists need to calculate stuff. Then you can sort of measure the effectiveness in a way. It is very vague. Because you have the same product in a way, which is very hard to gauge whether it is the same functionality
P1: Yes, i was thinking clicks per user per time. But I am not sure, the better the product is, the longer you hold your user, the more usable your product is the more people interact with this
R: The goal of this session is just to brainstorm as many ideas as possible. Because we will also do this session with other teams and we will try to see if there is more support for some measures, so you can just write down all of the things
P3: The number of clicks needed to perform a task, so that is a measure. Not a threshold or something, but it is a measure. Number of clicks, or number of page views, to complete a task.
P1: I would say that is a better one. My handwriting is actually terrible. This is actually very readable
P2: So usability, number clicks to complete a task, okay
P3: You can measure it in clicks you can also measure it in time
P1: I think it is about the same measure
R: I think they are very good, but we can still improve them in the sense that for one application it makes sense to have ten clicks, but for some it makes sense to only have two clicks, depending on the functionality. So how can we define a baseline? What should be the optimal result?
P1: Yes, so I would say you would compare this, well almost always, with yourself. So you would always say you take your own baseline. P0 basically, and then try to improve this. Instead of trying to compare this. Unless there is a very direct comparison, but I think that is always super difficult. For certain standardised options, like resetting a password, that is a very standardised thing that a lot of applications have, where you could put a number of clicks to it. So for certain standardised functions I would say just have some sort of maximum that says please fall within this maximum, do not take a million clicks if you send an email, an unsubscribe button should be one click. That is a very hard measure for usability. And that is an actual law I think as well, to have this being usable. I would say the same for those blind and hearing aid things, that they need to be as well. They need to be one or two clicks away, they can't be searching for that. I think for help of hearing, there is also a way to format your text. So that auto readers can focus on the main content first. So you define that in your application, that is also just a yes no thing basically, I would say
P3: Yes there are standards for that, I do not know how they are called
P1: Yes basically, it tries to read home, contact that, whatever first, and then it starts with the main page and then you have five minutes before that
R: We are talking a lot about the adherence of standard I hear. How would you measure this? Because there could be a human potentially involved in evaluating whether we actually

adhere to standards. Or how would you measure this? Because then it would be a subjective measure to ask someone to say hey
P1: I would say that, lets just take a hearing aided program, or a visual impaired program, because they cant see they need text to speech basically. How long does it take to get to the main content of the page in minutes. I would say that is a very objective measure of how long it does take for that program to find the mess you make
P3: Can it not be simpler, like for usability someone, blind people should be able to use the product, and then your usability goes up
P1: But at that point you have to find an actual blind person to test your program. But depending on how large your product is, that is very difficult basically
P2: Also kind of subjective
P3: It kind of depends on your requirements
P2: Also what does it mean to actually use the product? You can use eighty percent of the functionality, can you use it, can you score it in a way? So maybe time to speech, how long does it take to get to the extra information.
P3: You can have the measure but maybe it is the same argument, if it is your product you do not need it. For government sites it is obliged and also for commercial site that is not yet, so
P1: But it does not need to be required right, it is just more usable because some people are
P3: I think it can be simpler, not necessarily to measure it in milliseconds, or to say if it supports this, then it is more usable, if it supports text to speech it is more usable
P1: But the reason I mention this because I know someone who is visually impaired. A lot of things are usable, but the time it takes for it to be usable, there is like a big gap in it. Because those programs are kind of smart and they kind of see past it. But if you actually make sure your application works with it it is a lot better and faster
P3: You can drill down on it, because
P1: Especially, because this is such a hard category. How do you measure this for non visually, regular people is very difficult I think
P2: I think that is the thing we talked about at the start. Are there some standardised tasks, or any other functions we can think of that most applications use
P3: But the question is, how do you want to measure it. You can measure it using surveys, are you can measure it using number of clicks to perform tasks. That is the way to measure it
P2: Text editor, you can measure words or the time to something. But then what are the type
P1: I would say minimum click paths to standardized functions. Like how long are the minimum click paths for a standardized functions. Make a few functions that are in most applications and then you can actually make it as an integer, just get an integer out of it. That is okay or not okay. It is not to judge right, it is to measure.
P2: Have we captured that in those three yellow post-its, because we have talked a lot
P1: Yes, number of clicks could be
P3: To a standardized function, exactly
P1: Well, I do not know anything else
P3: Same
R: Let’s move onto reliability. With reliability being the degree to which a software product performs specified functions under specified conditions for a specified period of time
P3: Well, you can calculate uptime, very simple right, how long is the system up compared to for example your SLA or something. Even uptime itself would be a good measure I guess
P2: Mean time between failures?

P3: If we have three outages and how quickly you recover
P1: I think for API's there are standardised ways to measure uptime, for websites there are standardised ways to measure uptime. So those are very objective, and very valuable to have
P2: Do we miss something? Uptime, mean time and what was that you said P1?
P3: I think it falls under this, uptime or sort of api performance or some standardised protocols
P1: You can measure uptime on several things, I think the main things are basically if you have a website, how often can you reach the website, then you have how often can that website reach its own backend, in some way, that is also not hundred percent. And then you have for an API the same basically. How often can you reach an API? The API reach his own backend. Because it can send a 500 internal server error, but then you have a 100% uptime, but maybe not
P3: Maybe not really no
P2: And recovery time? Time to recover from failures? Is that something that we want under reliability? I think you have a reliable system if you can recover from failures
P1: I think it is super hard to measure
P3: So you can measure kind of like a clean deployment right. Suppose everything goes to hell, how long does it take for this to deploy again
P1: But for some things, if GitHub goes down
P2: Then you could argue data loss
P1: How is data loss?
P2: And that should be zero
P1: The data loss during down time, because that is what GitHub ran into. They could restore it, but they could go back six hours in time, but that loses million of hours in programming time
P2: And the resulting measure, that should be zero. And the recovery time?
P1: Recovery time I find difficult, like how long, that is sort of part of uptime right? The longer your recovery time, the lower your uptime will be
P3: That also depends on the complexity of the system, and also the data loss I feel. It is hard to quantify
P1: I would say it would basically be duration of downtime of one percentile downtime duration, the longest downtime duration. Like how long are they. What is the worse that could happen. What is the one percentile downtime duration?
P3: I am going to write down: what is the worst that can happen
P1: That is pretty objective, if all your downtime things, average duration can be important, but it is not really related to the quality of your product. Like ING probably goes down a couple of times per month for five minutes. But they hit the news every time they go down for a day. That is what they care about basically, more than, that is more qualitative. Last year they went down five or six times and that was a news article everytime it went down
P3: That was not a good time, that was very stressful.
P2: What else is related to this?
P1: I think this is very measurable. We got some pretty broad. I think this is by far the most measurable thing
P2: Dimensions, quantities, yes
P1: And I think the *inaudible* are spent in the software industry. We can maybe move on.

R: So looking at security, being the degree to which a product or system protects information and data so that persons, products and systems have the degree of data access appropriate to their types and levels of authorization.
P2: And how do you measure that?
R: How to measure that, yes
P3: Amount of data breaches, zero. Time to last data breach or something, but that is a very rough one
P2: You are measuring your end product
P3: The degree of data access appropriate
P1: I think there is an objective measure here in , there are several EU guidelines on security that you need to hit. I forgot the name. The reason we do security is because we are a *industry type* company and we have a hard requirements on that. That is just like a pass, fail check with a million parameters basically. A hundred parameters actually. The security controls implemented, that is what you can measure. I forgot the name, maybe you already know
P2: The nis two, the more frameworks. I can generalise it like security controls of common frameworks implemented
P1: I wouldnt call it frameworks, this is actually law, like you are breaking the law if you do not do it
P3: Compliance with the law sort of, is the measure.
P1: And it is becoming a lot more measurable. Ten years ago there was nothing measurable in this. But now, there is very specific rules on this.
P3: Oddly specific almost
P1: For certain sectors at least
P2: So you can actually make it, law zero one. You adhere or you do not. And you can make it like the number of controls you have implemented
P1: There are a lot of controls, and ofcourse there is a final thing. Do you hit all of them, yes or know. But then the data breach stuff is very hard to quantify
P3: Time to last data breach or something. Or amount of data breaches, but it is not, I mean, is it a measure that actually says something about how secure a system is
P2: It is saying how much secure because I measure the number of burglary attempts, or you say my house is secure because my locks are ISO I do not know what and I have cameras, surveillance et cetera. I think it is the last category we have to look into. Why is my product secure?
P1: One thing you can measure is basically, the amount of users you have lost any data for to outside forces compared to your userbase basically. Like what percentage of users have you lost. That is definitely a percentage
P2: It feels like we measure it when an incident has happened, and you want to make some kind of standard like when is my product save or when is my product usable. Okay, my product is usable because I have a very easy to handle for my door so you can walk in easily, or my doors open automatically so it is very usable
P1: You mention basically like beforehand, but you can also measure afterwards
P2: Yes, the number of breaches is also hard to quantify
R: We have fifteen more minutes for the discussion
P2: Okay, sorry
P1: Is it hard to quantify? A lot of people just know, we lost a hundred of thousands email addresses, we know how many email addresses we have. It is just. Security

P3: It is hard to quantify it beforehand, before you have lost the data from some people, it is hard to quantify how secure it is
P1: For the measure you can say, secure everything so you do not lose data, but it still happens
P3: Of course, no, it is just the name of the game. I think that is so hard to quantify security, and put a number on it
P1: Yes, so
P3: We talked about encryption for example, so do you set data encrypted
P1: How much of your data is encrypted?
P2: Standards or controls that you implemented, yes. And you can do a security test on a product, and you have a number of findings versus the number of. But then you are testing your product, you can
P1: I think the frameworks for law are already very broad and specific
P2: It should be expected to be in a control
P1: So that is already on the front
P2: And then you can say you test it, and then they have a measure. Because you have a test
P1: Are there security related incidents that are not data breaches?
P2: Yes, anything you can do to attack a system to make it unavailable that is not a data breach. Hijacked email site, that is not a data breach
P1: I would say unavailability due to security issues. Because you have unavailability due to whatever performance downtime, whatever. But you also have downtime due to security basically
P3: That is it for now I think
R: Let's move on to the second last part. Which is maintainability, which is the degree of effectiveness and efficiency to which a product or system can be modified by the intended maintainers
P1: Is it modify to add functionality, or modify to make sure it works
R: Both
P1: So, I think this would, adding features to having some amount of weeks until feature request goes from requested to implemented
P2: Yes, the cycle time or how do you call it
P3: Lifetime of a feature basically
P2: There is a name for it, cycle time
R: Cycle time or lead time perhaps?
P2: Lead time is one to ten, cycle time, lead time of feature. Of new feature
P1: I think you have this one and then the counterpart which is the lead time of bugfix, which is just the same but then for a bugfix basically
P3: I think also about for example, it is different, but think about for example onboarding, how long does it take for people to know what is going on, so they can maintain it. That is very vague. Also how many people work on it. It can still be very maintainable if a lot of people work on it.
R: Maybe a question that could help towards finding objective measure could be: how can you make your product more maintainable? What will you implement to make your product more maintainable
P2: Yes, that is good. More maintainable
P3: I would say codebase covered by tests sort of a percentage. So you have like a thousand lines of function code, and if it is a thousand lines covered by unit tests or

whatever. Percentage of code covered by tests, divided of course into different, not a hundred percent, but
P1: That is a very objective measure, tests
P2: Automation helps in maintainability. So the CI/CD stuff
P1: So how long to CI/CD pipeline lasts basically, how long does it take to implement a change. What is the deployment time? Let's say you have to make a real big change, how long and what is the time
P2: If you measure the one to one product, you can say is your whole process automated with CI/CD, then you have a more maintainable product. And that is more like a yes no then a kind of number
P3: Do you have CI/CD implemented?
P2: But CI/CD is such a broad term and I do not like it. Your whole process including testing is automated
P1: But everyone will say yes to it, but it is not true
P2: But we are living in a different world
P1: A lot of people will say yes to it, and there will be very different degrees between those what they mean with it
P2: In the past it was all manual testing, manual integration testing, manual acceptance testing, the deployments were manual. Manual descriptions
P1: But I agree that a yes no is important as well. But if it is yes it is still a wide spectrum of possibilities. But how do you measure where on the spectrum you are. There are certain maturity models here. The maturity you see, but that is also. I think that is an actual measure, maturity of, you have one to four, one to five. I would mention here something about break glass procedure. Like how solid is it, and how to quantify that, I find very difficult. It needs to be secure, it needs to be fast, it needs to be reviewed, it needs to be like it can cause a lot of issues if you have a hard break glass thing. But it can also solve a lot of big incidents
P3: So some sort of, ridiculous, but how many people have admin access to your product? And then basically you have breaking glass, zero, until it is necessary
P2: But I think it is more like
P3: Not sure, just thinking out loud. It is an interesting one, because we do not do that here, breaking glass
P1: They are working on it, but the issue is not like, there is issues with breaking glass basically. If it takes too long, or it takes not long enough, there is no check. But if there is not at all, if you just have an admin with a weekend shift and have to on its own fix a production database, that happens a lot
P3: It happens for sure, I can imagine
P2: What do you need if you want to maintain a product, what do you expect or what the project does need. I would argue that you need documentation, you would need to have it clear written, and code
P1: But documentation is also, it is not a yes no thing again
P2: I understand that
P1: I fully agree with everything basically but I am trying to come up with how do you measure this in a way. I think P3 was sort of on to something with how long does it take for new people to be able to maintain
P3: But it is so hard to quantify. I think mainly for maintainability, the amount of problems you have to fix manually, or that you can do using CI/CD for example. But it is also not really, I mean quantifiable in a way, you can actually count API calls in the console or something. It

would be. Well this is interesting as well because I had to do the whole restauration of the *platform name*, and I think for maintainability it is an integer. The number of manual steps you have to do to do a clean deployment. Because all these manual steps also add complexity
P1: But this is not really fair because I would say that google for example is probably very maintainable, and they won't really be able to do this. They will just, it is just too complicated
P2: This is what you see from the outside. And from the outside, you have already implemented a new feature. That is quick, or when it takes weeks everything has to be tested again, the code has to be. I think. This is what you can measure from the outside, objectively
R: Okay, due to the time, shall we move on to the last one. The last one is portability, which is the degree of effectiveness and efficiency to which a product or system can be transferred from one hardware or software operational or usage environment to another
P2: The number of backend data ports from the PC to XBOX or reverse
P1: So basically how vendor locked are you. Okay
P2: But an objective measure, is it the time to port from one system to another. If it is running on a docker container it is zero for example
P1: It is not zero
P2: Well, almost
P1: But if you have like a, of course you can say the time it takes to move it. But no one is going to move it until they have to. Of course at some point you can measure it, because you have to move it, and then you have a value, but no one is going to do this for the exercise of it probably
P2: Then it is how do you define portability, then it is almost the definition. My system is portable because it can be moved from one system to another without effort or recoding
P1: I would say one thing that is super valuable in this is that you have your hardware connectivity of your code separate from the functionality of your code. How do you measure this, I think is very difficult. But it is something that if you do not have this. Let's say you are an Amazon, you have some functions. You want to separate your S3 reader function from your model and then your S3 output function can be separate. If you have separated that, you can quite easily transfer your stuff to something else, like to Azure. But if you do not have it separate
P2: Separation of functional and
P1: I would say the amount of platform specific API calls in your functional code
P3: That is a real measure, that is a real one, yes. If you use this amount it is very specific yes
P1: If you have a boto call in your code you are screwed basically
P2: Yes, that is a separation yes
P1: It makes you super vendor locked
P3: It is not a joke anymore
P2: Platform specific code in functional code. Amount of platform specific code in functional code
P1: The other thing is basically, so how much of the platform related things do you actually orchestrate, or how much of your come together of your microservices is specific to the platform, or the thing you are working on
P2: Is it not the same? As platform specific in your functional, you do something functional and that should not

P1: I think this is really looking into the microservices self, but if you are looking at how your microservices are linked. You can do it through an EC2 that is running Apache Airflow, that would be movable. But if you are using Stepfunctions, then it is zero percent movable, or portable. But how do you, how much of your microservice orchestration is portable
P2: Make it very specific, microservice orchestration
P1: But microservice orchestration is something a lot of applications use
P3: But I can agree that is is some sort of virtual machine that you can carry basically anywhere, and is not platform specific. Probably azure has some sort of stepfunctions orchestration tool forsure
P1: A hundred percent
P3: But it is not transferable at all. But if you just have your docker containers, you can just spit them up
P2: But how would you make that product you are talking about more portable
P1: But I do not agree with what P3 said, you have two situations. You can have like an ECS that has a docker image on it. And if you have stepfunctions. You are still completely docker based. But if you have an EC2 machine with docker installed, and a separate EC2 machine with Apache Airflow installed, you can also orchestrate all sort of things, and it is much more portable. So microservice orchestration in this case really impacts portability
P3: Yes, good point
P1: But how to measure this
P3: Is there some sort of measure, like a ratio or coverage
P2: Can you state something in your architecture where you have an architecture of your system and you can measure or quantify it
P1: But how do you measure architecture
P2: Or some kind of quality attribute
R: We can think of the same question: what can we do to improve the portability of the system?
P2: So if you look at the architecture, if you have the wrong architecture then you are lost. So in that phase if you have a proper architecture which already takes into account portability, but my only question is then how to quantify that
P1: Maybe we can go back a step further than this, this assumes that this is a good quality to have, to make it portable. Well, a lot of companies, including our company made a very specific choice to make it a very low priority
P2: But that is for all aspects, you can also argue that this is not important for us, usability is less important than security for example
P3: That is a good point, when we think of *platform name*, this is something that for *platform name 2*, in a docker situation as well, but for *platform name*, we never think about it. When you ask how to make it more portable, we never think about it
P1: We made at the start at the architecture phase, I made the very conscious decision to make it zero percent portable
P2: But it is still an attribute, it is not a concern at the time
P1: But still the measure, R asks how would we make *platform name* more portable, that is difficult, but how to make *platform name 2* more portable, I really think it is removing the microservice orchestration outside a platform
P3: Then you have Apache Airflow and Kendra again
P1: But that is the way you want to improve it
P2: But if you want to abstract that, in kind of an abstract statement how would you move that. It is in your architecture then, in your architectural phase

P3: It is not a measure of anything basically. The amount of proprietary microservices some sort of open source, but you can really
P1: Even if you use lambdas for everything, which are proprietary microservices. If your orchestrator is not. I would say it is easier to move something that has everything in a lambda, but your orchestration not in AWS, then it is to move a system where your code is in Docker's, but your orchestration is in AWS
P2: I have a definition, it is more like a definition, but the number of major platforms you can run on without code modification, so you can port it. But it is a measure, but then if we can only run on AWS, without modifying the code
P1: But this will always be zero
P2: No, because you also have programs which run under Windows, and MacOs without code modification. And it depends on the context of course, for cloud you could argue Google Cloud, Azure, AWS, but for a desktop application you can argue the major OS's. WIndows, Linux, MacOS, so then you are portable in that area
P1: I find it very hard to come up with a measure like this, because an actual thing that would give me. I can say you that *plaform name* is less portable than *platform name 2*, and there are applications that are much more portable than that. But what makes it?
P2: If the *company name* app is portable, I would argue that the *company name* app is portable, if it can run without modification on Ios and Android
P1: But is that what you are looking for? For the end user to have multiple platforms on, but for the software itself?
R: It is about the software itself
P2: The same code can run on multiple platforms
P1: But the backend is still in AWS
P3: Exactly, can we move it to Google Cloud or Azure without moving it
R: I think what you mentioned on can the user run it on multiple systems is more related to compatibility
P3: I have no idea on this one, it is very hard
R: That is a research finding as well, I think we are almost at the end of the session, so thank you very much for all those ideas. I will make a picture of those and then I will end the recording