

R: Dan ben ik nu de recording gestart. Allereerst bedankt dat jullie mee willen doen aan deze sessie. Is het oke voor jullie als deze sessie opgenomen wordt?

P1, P2, P3, P4: Ja

R: Oke, bedankt. Deze sessie zal uit twee delen bestaan. Allereerst geef ik een korte introductie over het onderzoek en de standaard die we gaan bekijken. Daarna zullen we een discussie voeren en zullen jullie voornamelijk aan de slag gaan. Zoals eerder ook al gezegd het formele gedeelte. Ik doe onderzoek naar software product kwaliteit. Daarbij ben ik gaan kijken naar verschillende modellen uit de wetenschap en ben ik erachter gekomen dat er de meeste support is voor het model ISO 25010. Het is dus een ISO standaard. Het is een hiërarchisch model dat software product kwaliteit op deelt in drie niveaus. Dus op het top niveau hebben we de standaard, en het abstracte concept software product kwaliteit. Dan een niveau lager hebben we de quality characteristics zoals deze heten, en dat zijn functional suitability, performance efficiency, compatibility, maintainability, usability, reliability, portability en security. Het model heeft ook een derde niveau, sub karakteristieken, maar die zullen we niet bespreken tijdens deze sessie. We zijn er dus voornamelijk op gefocust om concrete metrics te vinden voor deze acht karakteristieken. Dus wat we nu gaan doen, ik ga een definitie voorlezen van een van de karakteristieken. Dus we beginnen links en daarna gaan we naar rechts. Dus jullie mogen iets dichterbij komen, zodat we ze allemaal kunnen zien dat we ze goed kunnen bekijken. En dan mogen jullie discussie voeren over hoe we mogelijk deze quality characteristic het best mogelijk objectief kunnen meten. Oke?

P1: Ja

R: Is het duidelijk voor jullie wat de taak is?

P2: Het gaat om objectieve metrics bepalen naar aanleiding van deze

R: Dat klopt helemaal, dus de procedure die we nu gaan volgen. Ik zal een definitie voorlezen, en jullie kunnen dan een discussie daarover voeren. Ik heb daar in ieder geval een pen en post-its neergelegd. Ook belangrijk inderdaad is dat er geen foute antwoorden zijn, dus bespreek gewoon alles wat er ter gedachte komt. Het is altijd nog mogelijk om met post-its te schrijven. Mochten jullie een metric bedenken en er later achter komen, de eerdere metric past beter hierbij, dan is het ook nog mogelijk om die te switchen. Yes?

P2: Kan je ook een metric bedenken die onder meerdere valt?

R: Zeker, als je dat denkt inderdaad dan maken we meerdere post its aan, dus we plakken hem dan eerst onder deze, en daarna onder die. Yes? Oke, de eerste karakteristiek is functional suitability, en de standaard definieert functional suitability als de "degree to which a product or system provides functions that meet stated and implied needs". Dus dan nu de vraag, dat zal voor de andere zeven karakteristieken ook zo zijn: hoe zouden we functional suitability kunnen meten? Als een definitie niet duidelijk is laat het ook gewoon weten dan kan ik iets verder exemplifien?

P2: Wat wordt bedoeld met functional suitability? Als in

P1: In het nederlands is het geschiktheid. Of de software geschikt is voor een doel

R: Dat klopt helemaal inderdaad

P1: Dus hoe bepalen wij als software engineers dat een product geschikt is voor een doel, voor een opdrachtgever, nouja

R: Ja, exact

P2: Ik zit meteen te denken aan een KPI voor een product, maar dat is dus heel afhankelijk

P1: Doen we dat ook echt?

P2: Nee dat doen we nu zeker niet, maar idealiter

P1: Maar de KPI's die we vaak hebben misschien rondom Scrum is werkwijze, hoe je tot je product komt zeg maar. Of het product is een dashboard en dan is het een KPI en de vraag om een software product te leveren is breder dan dat

P2: Maar het gaat er dus wel om dat het zijn doel vervult. Je wilt dan iets hebben, zeg maar, wat het doel vertegenwoordigt op die manier. Maar uberhaupt ik weet niet of je daar wilt nadenken over true false of dat het een getalletje is

P1: Nee, ik denk dat wat die vooral wilt hebben, hoe bepalen wij dat nu

R: Ja de vraag is eigenlijk, wat voor getallen zouden we hier aan kunnen hangen dat als we die getallen gaan meten dat we kunnen meten, is een product functioneel suitable of is die dat niet. Dus er is inderdaad ook een verschil in de mate van functional suitability

P1: Als ik de andere kopjes bekijk valt die niet onder de andere in ieder geval, hij valt echt onder de linker. Ik zit te denken, via Scrum werken bij de klant, wat bijna iedereen van ons wel doet, zeker in de IT, een bepaalde geschiktheid idealiter, niet altijd, maar is een sprint review houden. Een klant komt, stelt vragen, is het er mee eens of niet. Als het niet geschikt is dan zegt die het kan dit niet, of het doet dat niet. Dus het gaat meer om de verwachting afstemmen met de klant. Naar aanleiding van, kortom, oplevering. Ik weet niet of je dat in een getal kan stoppen

P2: Maar dan zit je wel in de objectiviteit van de stakeholder die op dat moment aan het spreken is

P3: Maar dat is toch niet meetbaar?

P2: Ja, dan krijgen we inderdaad een objectieve mening van de persoon die op dat moment praat. Ik zit zelf te denken, als het ook over meningen gaat, vanuit mijn AI kant, kunnen we die large language models het kunnen bepalen. Die zit wat objectiever in wat het doel is. Dus je beschrijft het doel, laat achteraf de code bekijken, ook als input, en je vraagt aan hem: sluit het resultaat aan bij de beschrijving die de opdrachtgever heeft gegeven

P1: Nou, ja, dat is ook iets wat we wel doen, of kunnen doen inderdaad

P3: Performance meten ook, dat heeft er ook mee te maken natuurlijk

P4: Heeft het sowieso niet te maken met wat het resultaat voor een format is? Dus als het resultaat een getal is, het resultaat een gevoel

P2: Maar het gevoel wil je dus weghalen omdat het gaat over objectiviteit

P4: Nou oke, dan moet je kijken hoe kan je dat desentisizen. Je kan een likert scale gebruiken: hoe boos ben je op een schaal van een tot tien? Dan kan je er een getal aan hangen, ik zeg maar wat

P1: Maar wat bijvoorbeeld ook zo is dat we idealiter, in Scrum heb je in je user story staan: acceptance criteria. Dus voldoet het aan de acceptance criteria? Maarja, of we dat dan echt vaak doen

R: Het gaat er inderdaad om, wat is een potentiële manier om dit te doen, dus of het momenteel gedaan wordt of niet

P1: In ieder geval kunnen we het opschrijven, of het dan objectief of niet. Een review is iets wat we normaal gesproken doen, iemand het ermee oneens?

P2: Nee, zeker

P3: Ja

P1: In ieder geval dat noemen, en dan het andere voorbeeld

P2: Dat van large language models

R: Dus dit is de numerieke waarde die hier uit komt rollen

P1: Het is geen numerieke waarde, het is een meeting

R: Maar is de waarde die je dan wilt meten, er wordt wel of niet een sprint review gehouden, of wat zou de output hiervan zijn?

P3: Je zou een sprint comparison kunnen doen. Dus met supervised learning, ik verwacht dat dit eruit komt. Is het resultaat hetzelfde

P1: Oke en als je dan een dataset gaat vergelijken

P2: Dat je code, dat klinkt alsof je gewoon aan het testen bent. Use cases aan het testen bent

P4: Ik zit even te denken, de need is als we terug gaan naar het voorbeeld van de auto: hij gaat, kan ik dit goed zeggen, de auto wordt groen gespoten, we verwachten een groene auto. We kijken naar het resultaat: is het een groene auto?

P2: Ja \*inaudible\*. Dat klopt, maar dan kom je uit op een soort code coverage qua testing

P4: Niet gewoon unit tests?

P2: Ja, unit tests. Integration tests, al dat soort dingen. En dan wil je zo'n hoge code coverage hebben

P4: Dan zou ik misschien zeggen testen

P1: Testen inderdaad, maar is dat hetzelfde als geschiktheid? Iets kan perfect getest zijn maar niet overeenkomen. Denk aan een fiets bouwen, en dan vierkante wielen krijgen, of met zes wielen?

P2: Je vangt het niet af omdat je subjectiviteit in het testen hebt

P3: Maar het lijkt mij dat je dan kan meten of het de implied behoefte, ik bedoel nee is ook een antwoord toch?

P1: En het hangt van het testen

P2: De kwaliteit van de test af

P1: Ja, en wat je wilt, en wat je hebt getest. Dus in dat opzicht kunnen we best testen zeggen, ik zie het hier ook niet onder vallen. Tests, hoe noem je dat dan?

P3: Test framework?

P1: Metrieken ofzo, test?

P2: Slagingspercentage van tests?

P1: Test slagingspercentage, oke

R: Ik hou af en toe de tijd in de gaten, dat we ook over de andere dingen kunnen praten. Laten we hem op twee minuten timeboxen. Nog twee minuten voor functional suitability

P1: Acceptatietesten hebben we nog niet opgeschreven. Large language models, LLM gewoon?

P2: Ja, LLM

P1: Maar wat is de metriek dan?

P2: Ja die kunnen de essence van bijvoorbeeld is dit mooier: mooier is een heel erg vaag concept. Maar toch als je zegt, je hebt een plaatje: maak deze mooier. Zie je dat die daarna een image genereert. Dus large language models hebben de capaciteit om een vaag concept beter te begrijpen. Dus je hebt een language model, en je prompt engineer er een vraag aan, voldoet het aan het doel? Dan komt daar een true false metric uit: door de code en het document van functional suitability. Als je daar een document van hebt kan je het bij een large language model stoppen en dan kan die zeggen: ja of nee

P1: Maar stel dan: zegt de LLM dat het aan het doel voldoet?

P3: Dat is gewoon een true of false dan, of een rating

P1: Nou je kan ook zeggen, dat hangt van je context af, je kan ook zeggen van: tussen nul en tien, in hoeverre komt het overeen met je doel. En dan komt die letterlijk met een getalletje

R: Zou je dit eenmaal per product doen? Zou je dit voor elke functionaliteit doen? Wat voor niveau van granulariteit zou je dit het beste kunnen doen?

P2: Per feature denk ik? Het totale pakket

P3: Per feature, bij ons is feature misschien wat specifiek afgericht. Als een feature af is dan is die eigenlijk al af en zou de klant al functionele testen gedaan moeten hebben. Bij ons is het, ik weet niet, het kan misschien bij jullie, ik weet niet hoe het volgens Scrum, wat de juiste benaming daarvoor is. Per increment zou ik misschien zeggen, per sprint? En als er niks is, dan is er niks. Maar als er wel iets is dan kan het wel functioneel getest worden met deze meetwaarde

P1: Maar ik denk dat het vooral wel over alle dingen die we zeggen is, je kan het per kleine dingen testen, per item, per sprint. Maar je kan het ook per feature, dat geldt met al van deze dingen ook

P2: Misschien wil je het overal op loslaten, als je toch de optie hebt. Op elke unit die je kan bedenken

P1: Eentje die we nog niet hebben, en je bij sommige software wel hebt is in de praktijk testen. User acceptance test ofzo, dat soort dingen. Is ook iets om te meten of het geschikt is

P2: Ja, ik zit dan te denken, meer aan het gedrag van de gebruiker zelf wat je aan het meten bent, en dan stel je hebt een website, dan kan je met hotjar bekijken waar wordt die veel gebruikt. Naar aanleiding van de data die je hebt, dan komen we een beetje bij het eerste wat ik zie, als je losse metrics opstelt, doelen, dan kan je data capturen die je vraag beantwoord. Maar dat is denk ik niet iets wat we kunnen opschrijven, het wordt elke keer een losse onderzoeksvraag, met losse data

P1: User tests, systeem tests? Of is dat net de verkeerde. Ik zit even te denken welke het normaal is. Het gaat om de niet technische tests

P2: User behaviour monitoren, naar aanleiding van het doel wat je definieert

R: Ik denk dat aan de hand van de discussie de metric wel duidelijk is, en hoe we hem dan noemen is iets waar ik mij op los kan laten

P1: Volgensmij is het normaal user acceptance test, hoe dit heet

R: Oke laten we in verband met de tijd doorgaan naar de volgende. Iedereen akkoord? Nog slot conclusies?

P2: Nee

R: Dan gaan we door naar de volgende: dat is performance efficiency. En de standaard definieert performance efficiency als de "performance relative to the amount of resources used under stated conditions". Dus hoe goed doet een product het in een bepaalde omgeving als die bepaalde resources heeft

P4: Loss function? I guess toch

P2: Fair, ik zat het zelf eerder te vertalen naar het volgende: als er verschillende resources wegvallen, gaat de performance dan zwaar omlaag. Is dat wat de vraag betekent?

R: Ja zeker

P2: Dan gaat het eigenlijk over, hoe heet het, daar heb je een woord voor, dus ga maar door

P1: Maar het gaat toch wel om de performance efficiency? Gaat het om het maken van de software of het met veel of weinig mensen?

R: Ik kan natuurlijk geen voorbeeld geven in deze context

P1: Dan kijk ik ook even naar de anderen, hoe lezen jullie de vraag? Het kan om de performance van de software zelf gaan, of het werkt, of hoe we het gemaakt hebben?

R: Het gaat specifiek om het product. We proberen de kwaliteit van het product te meten

P2: Ben je bekend met de term anti fragile?

P1: Anti fragile?

P2: Ja, als iets fragiel is gaat het heel makkelijk kapot breken. En anti fragile is een concept dat hoe meer pressure of druk je erop uit oefent, hoe sterker wordt het. Dus als ik kijk naar

het aantal resources dat erbij betrokken zijn gaat het eigenlijk over de fragiliteit van je software, klopt dat? In hoeverre dat je systeem inperkt

R: Ja, dat kan

P1: Ja, je kan natuurlijk zeggen, als je veel resources hebt dan heb je veel snijverliezen, overdracht dingen, waardoor het soms niet goed werkt, de software. Dus je kan zeggen hoe minder mensen eraan werken. Als er een iemand aan werkt dan werkt alles samen, omdat diegene ook verantwoordelijk is. Maar het kan ook een beetje in een andere hoek zitten

P2: Als het gaat over is een product afhankelijk van die ene server die er staat en die twee toolings die we gebruiken, als die uitvallen blijft je systeem dan overeind

R: Ik denk de vraag: blijft het overeind of blijft het niet overeind, dat is meer een reliability vraagstuk, performance efficiency is net een vraag binnen het kader dat jij niet hebt geschetst: hoe goed blijft het overeind?

P2: Oke, precies, de performance die erbij hoort

P1: Over de resources, dus niet de personen

P3: Stress test

P2: Dat soort dingen wil je doen, en als je kijkt hoe goed blijft de performance. Stress test komt het op neer. Mag opgeschreven worden. En iets wat in het algemeen heel erg werkt is dat je een ecosysteem hebt van vervangbaarheid. Dus als iets wegvalt dat je weet: ik heb nog een alternatief. Als een component wegvalt je niet in een keer een error krijgt

R: Zou dat niet reliability zijn? Ik denk omdat ze best wel met elkaar intertwined zijn dat het belangrijk is om reliability ook voor te lezen. Reliability is de "degree to which a product or system performs specified functions under specified conditions for a specified period of time". Dus reliability gaat vooral over: is iets functioneel onder bepaalde conditie: ja of nee. En performance efficiency gaat over hoe goed het performed als het een bepaalde hoeveelheid resources gebruikt

P1: Ik kwam net ook onder wat hier speelt: amount of resources dacht ik al. Als het een systeem is, of het veel ontkoppelt is. Als je veel connectie hebt gaat de performance omlaag, maar niet alleen de snelheid maar ook de betrouwbaarheid

P3: Want afhankelijk van andere systemen, gaat ook over reliability natuurlijk

R: We kunnen er hier ook een aantal bedenken en als we reliability bespreken gewoon nog wat bewegen

P1: Dan denk ik, de mate van koppelen of ontkoppelen van systemen

R: Hoe zou je dat concreet meten? Dat als ik nu naar een product zou kijken, wat zou deze waarde zijn?

P1: Aantal verbindingen tussen de systemen van je product

P3: Ik denk dan voornamelijk aan de betrouwbaarheid, dat valt natuurlijk ook onder reliability, maar de snelheid van het product

P4: Efficiency misschien?

P3: Efficiency ja, ik heb vaak genoeg met dashboards meegemaakt dat je een dashboard opent en drie uur moet wachten. Dat performed gewoon niet. Dat is hoe ik de performance zelf zou zien

P1: Dat heeft toch ook met die ontkoppeling te maken?

P3: Ja uiteindelijk wel, maar op het moment zelf is een performance test voor de gebruiker wat minder. Het gaat er ook over hoeveel mensen tegelijkertijd in het systeem zitten

P2: Maar qua metric, time to load?

P3: En als je het vertaald naar website

P2: Maar ook met dashboards

P3: Ja, maar ook met websites, hoe heet het ook alweer. Er is ook zo'n test die je kan doen, ik ben vergeten hoe het heet, en dan kan je zien hoe alles snel inlaad

R: Volgens mij bedoel je de time to interactive. Dus we hebben voor nu toe voor performance efficiency: stress testing, de mate van coupling, en de time to load. Ik denk dat we heel goed daar het stukje performance in vatten, maar we hebben nog niet heel erg gekeken naar de resources. Wat houdt dat in, en welke resources kunnen performance beïnvloeden?

P2: Als ik dan denk aan een dataset \*inaudible\*, dat is een resource die je nodig hebt voor een resource

P3: DWU of wat dan ook

P4: Structuur van de nodes

P1: Dan ga je het heel technisch maken

P2: Of de, hoe heet dat nou, de verhouding tussen netwerk of bandbreedte, ten opzichte van de size van de dataset. Ik weet alleen niet dat verhoud dat de tijd die je meet

P1: Under stated conditions, dan ga ik ervanuit dat gegeven een bepaalde netwerk grootte, gegeven een hoeveelheid processoren, gegeven een bepaalde grootte

P4: Hoe de data is opgeslagen

P3: En ook hoeveel het is, met aggregaties kan je natuurlijk heel veel doen. Nu gaat het een beetje technisch worden, dan heb ik wel iets van, op het moment dat je een hele grote dataset hebt kan je het wel binnen een seconde inladen op het moment dat je alle aggregaties goed hebt staan en alle dingen goed hebt ingezet. Maar hoe kan je dat weten? Dat is wel een beetje lastig. Ja snelheid, maar dat is niet een goede metric

P1: Dataset is dan eentje van die sources

P3: Ja hoeveelheid, ten opzichte van die laadsnelheid misschien. Time to load

R: Om hem te timeboxen, nog twee minuten voor deze

P2: Kunnen we nog andere dingen bedenken

R: Als de twee minuten niet nodig zijn kunnen we altijd door naar de volgende

P2: Ik denk dat we door kunnen naar de volgende

P1: Time to load, maar dat is meestal de eerste keer, of bedoel je hem ook tussendoor. Dus je kan hebben wanneer je dashboard opent. Maar het kan ook, je klikt ergens op en hoelang dat dan duurt. Kun je ook time to load noemen maar je kan ook zeggen, mate van interactiviteit

P2: Interactie, en openen, opstarten

R: Oke

P1: En gewoon de tijd kun je nog apart erbij schrijven

P2: Hier staat time to load, dat bedoel ik ermee

P1: Ik zit te denken, dashboards, maar ook datasets, als je een dataset maakt

R: Oke, de volgende quality characteristic is compatibility, en compatibility is de "degree to which a product or system can exchange information with other products or systems". Dus dit gaat specifiek over

P2: Dus stel je hebt diverse API's, om dan alle data eruit te krijgen

R: Het is belangrijk, wat in andere focus groups ookal ter sprake kwam, dat compatibiltiy soms een beetje een dubbel zinning concept is, en mensen zagen hem eerder als: hoe compatible is het met verschillende platformen, maar het gaat specifiek om het exchangen van informatie

P1: Of het een bekend data formaat is, hoe je dat in een metriek moet zetten?

P3: Is bekend, een of nul

P2: Nou, gewoon misschien

P1: Is het een gedefinieerd uitwisselingsformaat? Je kan ook zelf iets creëren, dan is het sowieso niet compatible

P3: Erkend misschien ook?

P1: In ieder geval bestandsformaat, bestandsstructuur moet ik het noemen. Dat is echt belangrijk, dan heb je er wel verschillende vormen voor: batch, stream, events, maar dat is misschien. File exchange format, maar je kan het breder trekken, data format

P2: Want in principe wil je nooit een getal op zichzelf laten staan, altijd in verhouding met iets anders

R: Iets wat in een vorige focus group ter sprake kwam, toen liepen mensen ook tegen deze issue aan van wat moet eigenlijk een baseline zijn. Als je er geen kan bedenken kan het ook zo zijn dat het puur zinvol is om hem te meten in relatie tot zichzelf, en dan hetzelfde product in een verschillende iteratie ervan. Dus als je bijvoorbeeld zegt: we meten hem nu, we meten hem over een maand. Dus als er geen sensible baseline te vinden is, probeer er geen sensible baseline aan te hangen

P2: Dan kan je zeggen het aantal verschillende formats die je aanbiedt, dus is het een .csv, .parquet, .pickle, . weet ik veel

P3: Dot net

P2: Dot net: Is dat ook wat jouw idee bedoelde P1?

P1: Ja klopt, maar ik had het ook gecombineerd met of het een bekend formaat is. Maar ik dacht ook aan aantal standaard formaten. Dus misschien het aantal standaard formaten, en het aantal niet standaard formaten

P2: Ja fair, we draaien hem om. Aantal niet standaard formaten. Des te hoger het getal is, des te niet compatible. Standaard formaten. Die kun je ook nog met verhouding tot elkaar zien

P1: Aantal veranderingen, aantal velden dat verandert. Ik denk nu een laagje dieper. We hebben er een data formaat, en over een maand, wat een van jullie zeggen nu van een maand later. Stel dat het tien velden waren en weer tien velden, dan is het compatible met vorige keer. Maar stel dat er een veld weg gaat, of veld bij gaat, is het al minder compatible. Als er veel meer verandert is het nog minder compatible

P2: Dus hoeveelheid interacties met dataset?

P1: Nee, aantal velden dat verandert is. Als je tijd bekijkt, tussen ontwikkelaars bijvoorbeeld. Of tussen ons en de leveranciers

P4: Wat heeft dat met compatibility te maken?

P1: Stel de overheid levert een formaat op, en over een maand, of over een jaar heeft het een ander formaat gekregen. Dat is het toch niet meer compatible. En dan gaat het erom hoeveel er verandert is

P2: Dus dat zou dan zijn, de duur dat iets niet verandert is

P1: Ja, als je het een niveau hoger bekijkt

P3: Of hoe vaak

P1: Dat vind ik ook wel een hele leuke, want elke keer als het verandert is. Als het heel vaak verandert is, begin je net zo goed opnieuw

P2: Dan krijg je daarna weer een nieuwe standaard, en daarna weer een nieuwe standaard

P1: Aantal keer dat het formaat verandert

P2: Ik zou dan niet het formaat zeggen, maar met formaat zat ik te denken aan de manier waar je het opslaat, en dit is meer onderliggend, de datastructuur. Aantal keer dat de datastructuur verandert

P3: Moet er ook compatibility tussen producten in? Of tussen operating systems en dergelijke

P2: En opzich, stel je hebt. Sorry ik praat door je heen

P3: Nee maakt niet uit. Ik vroeg alleen of we daar ook over na moeten denken

R: Ja zeker, het is “de degree to which a product or system can exchange information with other products or systems”, dus dat is inderdaad ook een goede om mee te nemen

P3: Of het tussen operating systems ook gewoon exchangeable is. Kan je data van een Apple naar een Windows sturen, of specifieker nog van systeem A naar systeem B waar de ene JSON en de andere XML is

P2: Ja

P1: Het aantal supported operating systems

R: Maar gaat het aantal operating systems ook over de informatie exchange? Of wat bedoelen we hier exact mee

P2: Wat P3 bedoelde is, je hebt een bepaald bestand dat kenmerkend is voor, laten we zeggen, word. Dat is dezelfde data, en je kan hem niet openen op Android als je niet de bijpassende OS hebt

R: Dus het aantal supported operating systems waarmee je data kan delen

P3: En het was vroeger ook zo dat als je Windows word had en je opent het op Mac word dan was het ook weer gelijk, tegenwoordig is het wel beter, maar niet compatible

P1: En dat komt dus onder andere omdat dat formaat, de andere opmerking die ik maakte, het formaat strakker is

P2: Hoe zitten we qua tijd?

R: We hebben nog 1 a 2 minuten voor deze, en dan moeten we naar de volgende

P1: Compatibiliteit, dat is natuurlijk wel een belangrijke

R: Altijd als je vast loopt in het nadenken kan je jezelf de vraag stellen: wat kan je doen om je product beter compatible te maken. En de acties die daaruit komen zou dan een metric kunnen zijn

P2: Ik zit nu anders te denken, maar dan komen we bij maintainability, want hoe meer compatible je bent met andere systemen qua informatieoverdracht, des te moeilijker het wordt te onderhouden

P3: Of je moet het juist heel universeel maken, het product. Dus dan ga je het hebben over die standaarden. Je kan zeggen ik exporteer alleen JSON klaar, punt, of wat dan ook

P2: Maar dat staat dan weer haaks op het aantal formats, die je wilt supporten

P3: Maar ook weer niet, het licht eraan of je het wilt, ja of nee

P2: Dus dan moet je met de hele industrie, de hele sector afspreken, we gebruiken alleen dit format, net zoals csv's. Csv's zijn het meest irritante format wat je kan hebben qua size, maar toch gebruiken we het overal

P3: Ik vind XML zelf vervelender, maar ja

P2: Ja fair, JSON vind ik heel chill

P1: Hebben we nog een andere manier van compatibiliteit

P4: Nested, nested, nested, nested

P2: Ja, dat hoort erbij

P1: Nested, dat is complexiteit. Dus hoe complexer, hoe minder compatible? Of is dat een aanname

P2: Ja zeker, hoe meer unstructured het format wordt, des te minder. Nouja met machine learning kan je tegenwoordig ook weer veel. Maar we kunnen het opschrijven, in theorie. Gewoon de data complexiteit op zich zelf. Hoeveel categorieën heb je, hoeveel datastructuren heb je, dat heeft ook een naam. Of het een getalletje is, zijn het woorden. Als het allemaal losse strings zijn in een ongestructureerd bestandsformat

P1: Een dieper dan exchange formats, dan kijk je op veldniveau. Type velden



R: Dus hoe kan je dit concreet meten?

P2: Ik denk dan het aantal van elk type dat je hebt. Bijvoorbeeld bij zo'n string, als het letterlijk alles kan zijn is het heel lastig om daar compatibiliteit mee te krijgen

P1: Of juist heel makkelijk, want strings is wel altijd gelijk. Maar als je een heel specifiek formaat hebt in Oracle Blobs, dan kan je het heel moeilijk lezen in een Microsoft omgeving. Maar toch denk ik dat het goed is om het te hebben over complexiteit, dus aantal, wat heb je opgeschreven?

P2: Type velden, aantal per type. Is dat een? Next

R: Yes, de volgende is maintainability, en maintainability is de "degree of effectiveness and efficiency to which a product or system can be modified by the intended maintainers"

P2: Ik zou zeggen de inverse van compatibility, gewoon alle metrics die je hebt met compatibility, de inverse van pakken, en als je helemaal niet breed denkt. Je support maar een line, een ding, lekker maintainable. Want eigenlijk is hoe kleiner en dunner je het houdt, des te beter maintainable. En hoe meer je aanbiedt in de breedte, des te

P1: Ja, maar dat is nog geen metric denk ik. Maar hoe kunnen we dat omzetten in een getal dan uiteindelijk in een vraag. Het aantal veranderingen die al succesvol zijn uitgevoerd, succesvol is dan belangrijk. Als je vaak dingen verandert hebt, en het was succesvol, het kan ermee werken, of je kan een aanpassing doen. Dan is de kans dat als je nog een keer een verandering doet. Wat in feite maintainability is, is een grotere kans. Als je het net klaar hebt, je hebt nog nooit iets aangepast. Over het algemeen, als je dan iets aanpast, moet je een stukje terug gaan

P2: Dus bijvoorbeeld de hoeveelheid Git commits die je kan plaatsen in een dag

P1: En toch ook, systemen die langer bestaan zijn qua kwaliteit betrouwbaarder. Ze zijn misschien lastiger met nieuwe mensen te doen want die kennen de technologie niet meer. Denk aan de COBOL discussie die deze week losbarst. Volgens mij zo'n beetje voor de X honderste keer. Het is niet zo dat wordt gezegd, die systemen draaien niet meer. Hooguit ze kunnen niet meer wat we nu willen, dat is iets wat in die eerste groep zit, functional suitability

P2: Wat mij betreft staan die twee haaks op elkaar, in de meeste gevallen, niet altijd

P1: Ben ik het niet helemaal, niet 180 graden. Ze hebben wel een relatie, en ze staan niet parallel van elkaar, daar ben ik het ook wel mee eens. Ik zou eerder 90 graden zeggen

P2: Ja fair

P1: Maar ik bedoel dus maintainability, het aantal verandering die succesvol zijn uitgevoerd

P2: Het aantal code veranderingen die productie hebben gehaald in een bepaalde tijd

P1: Kan ook, daarom dacht ik ook. Ik zeg gewoon succesvol. Hoe definieer je succes dan. Je hebt ook bagger systemen

P3: Of hele oude systemen die niet meer maintainable zijn eigenlijk. Of systemen die je nu maakt met een groep consultants, en over vijf jaar is de groep consultants weg en dan moeten zijn het maintainen maar dan kunnen ze niet doen omdat ze geen idee hebben waarom het moet

P2: Dus die domain knowledge zeg maar, de busfactor, de welbekende busfactor

P3: En de readability dan, de logische van een product

R: Hoe kan je dat inderdaad expressen

P3: Code quality zelfs nog eventueel

P2: Ja code quality, daar praten we over

P1: Aantal fouten, als je het echt objectief bekijkt er zijn van die meetinstituten in Amerika en je zegt, nouja jullie hebben drie komma zoveel code per duizend regels verkeerd, die zijn er. En hoe meer fouten erin, hoe minder onderhoudbaar.

Het is natuurlijk niet een op een relatie.

P3: Je hebt natuurlijk ook weer, als je een bepaald software systeem upgrade van, dit is deprecated maar dat soort dingen daarvan dan is het niet meer maintainable

P1: Dus dat heeft ook te maken met wat ik net zei over van Cobol van hoe actueel is het.

Hoeveel jaar geleden de code, dat is een getal. Hoeveel jaar geleden programmeertaal

P2: Je hebt algemene open source, GitHub, daar kan je heel makkelijk uithalen hoe vaak het gebruikt wordt. En dat kan je over de tijd plotten. En als je kijkt, hoeveel van mijn onderliggende libraries worden in andere projecten ook veel gebruikt, dus dan pak je de open data source van het gebruik van die libraries

P1: En wat is daar dan de meeteenheid van? Om tot een getal te komen?

P2: Dan zou je GitHub, GitLab, DevOps, alles waar repo's in staan, waar publiekelijk de libraries in staan. Dat is iets waar ook jaarlijks rapporten van komen

P1: Het aantal ontwikkelaars die kennis en ervaring hebben van de programmeertaal. Hoe meer dat er zijn, hoe beter het onderhoudbaar is

P3: Om het in aantal te noemen is misschien wel beter, maar eigenlijk is de stelling van wordt het nog veel gebruikt?

P2: Dus je wil bekijken bij de nieuwe repositories die er zijn, als er iets geupload is

P3: Een tijd geleden was Angular helemaal de shit, maar tegenwoordig gebruikt niemand het meer, bijvoorbeeld

P2: Ja, dus je kan gewoon kijken naar de trend. De trends die publiekelijk beschikbaar zijn op websites zoals GitHub. In hoeverre gebruikt mijn techstack

P3: Om eigenlijk ook een soort van, er is te zien welke programmeertaal jij gebruikt en per jaar gaat evalueren hoe die staat in de markt

P1: Ik heb hem opgeschreven en gecombineerd met wat jullie net zeiden: het aantal ontwikkelaars met kennis, ervaring en gebruik maken van de programmeertaal of softwaretool achter het product. En dan heb ik bewust maar even kennis, ervaring en gebruik maken samen. De ene keer is het misschien meer kennis, de andere keer is het misschien meer wat gebruikt men al

P2: En dan kan je dat meten via publieke data die beschikbaar is. Want daardoor herken je de trends

P1: Het aantal fouten in je code die er bestaan

P2: Ja dat soort dingen ook, sure

P4: Per aggregatie

R: En wat wordt dan geconsidered als een fout?

P1: Ja dat is natuurlijk wel een dingetje. Gewoon een programmeerfout bedoel ik even. Je hebt natuurlijk ook functionele fouten

R: Code smells

P2: Bad code smells. Dus je hebt er meer, dit is maar een deel, maar het is in ieder geval een deel.

R: In verband met de tijd, zullen we nog twee minuten proberen deze te doen, en dan door naar de volgende

P2: Voor de opname is het handig om niet allemaal door elkaar te praten

P1: Wat willen jullie zeggen

P3: We hadden een side discussion. Het was wel inhoudelijk trouwens. Ik denk niet dat je het kan onderscheiden in je recording

R: Oke, wat hadden jullie gezegd?

P3: We zeiden eigenlijk een beetje hetzelfde. Per regio is het ook belangrijk om te kijken welke programmeertaal. Ik zei tegen P4 toen in Amerika toen ik daar nog studeerde was R helemaal de shit, en daar gebruikten ze helemaal geen Python, terwijl in Nederland Python

heel populair was en R eigenlijk helemaal niet gebruikt werd, behalve door statistieken en universiteiten

P2: Of in Nederland is Azure echt de king in Cloud en in Amerika is het AWS

P4: Juist, juist. En waar Google Cloud eigenlijk P3? Duitsland?

P3: Ik weet het niet, misschien Rusland ofzo

P4: Daar hebben ze vast hun eigen Cloud

P3: Ja vast

P4: Ali baba cloud

P2: Gezien de tijd, opgeschreven: de mate of het aantal van de modulaire eenheden opzet van het product

P1: Dus de modulariteit

P2: En dat kan je meten. Alleen ja, hij zou net even wat beter opgeschreven worden

R: Dat kan ik doen bij het verwerken van de resultaten. Oke, laten we doorgaan naar usability, en usability is de "degree to which a product or system can be used by specified users to achieve goals with effectiveness, efficiency and satisfaction"

P2: Nouja, bijvoorbeeld time on screen, of screen time. De hoeveelheid tijd dat je in een applicatie in zit. Des te langer je erin zit, des te blijer ben je, toch?

P1: Er zijn mensen die klagen, het is zo langzaam ik zit al een uur in de applicatie, ik bedoel maar twee minuten.

P4: Maar dat zou je weer kunnen afvangen met kijken waar je muishover is. Als je inderdaad heel lang in de applicatie zit maar je muis beweegt niet, dan zou het te maken kunnen hebben met de loading time. Maar dat is een aanname

P2: Maar opzich zouden we, als je aan systemen denkt waar je een videoclip aan het bekijken bent, dan ga je niet de hele tijd je muis bewegen, maar je bent wel engaged

P1: Ik ben het ook wel met jullie eens, we hebben meerdere metrieken. Eentje is dan gewoon hoelang je gebruik maakt van de applicatie. Per dag, per maand. Per maand denk ik toch. Het kan zijn dat de ene metriek tegen werkt, en dat is mijn voorbeeld

R: Je kan natuurlijk, het idee is dat we zoveel mogelijk metrieken bedenken, en dat je uiteindelijk kan bedenken, welke maakt sense voor je product en welke niet. Bijvoorbeeld bij het invoeren van je belasting, dat wil je zo snel mogelijk doen, dus daar zou het niet sense maken om deze metriek te gebruiken, maar bij een videogame weer wel

P2: Of bij een pornowebsite zeg maar, de pagina waar je exit, dat is het beste clipje. Het zit in het onderzoek toch

P4: In principe de sky is de limit toch, want je zou ook facial recognition kunnen doen. Kijken of je iemand ziet lachen. Stel je voor dat je kan zien hoe iemands gezicht is, en je kan het koppelen aan emoties. Dan zou je ook brain activity kunnen pakken.

R: Het kan ook zo zijn dat je voor een groot deel van de applicaties je deze metriek niet zou pakken, maar het is nog steeds een useful metriek voor een bepaalde applicatie

P4: Dan zou je echt moeten hebben, disclaimer je webcam gaat aan

P2: Het is dan de vraag of je dat zou willen. Maar in theorie zijn het hele goede metrics

P1: Ik heb nu, via facial recognition herkennen hoe vaak gebruiker blij zijn tijdens het gebruik

P4: Of gewoon emotie herkenning, als ze boos zijn dan weet je het ook

P2: Op het vlak van user satisfaction. Mijn allereerste opdracht bij \*naam organisatie\* ging daarom. Het ging erom om inzichtelijk te krijgen of de klant gelukkig is met de applicatie. En daar kwamen we uiteindelijk op de metric van de dagelijkse gebruiker. En dan ging het niet erom, hoeveel gebruikers heb ik per dag, maar ging het er om: ben ik een gebruiker die dagelijks inlogt? En dagelijks, elke dag van de week, de applicatie gebruikt. Dan heb je wel het probleem waar ik tegenaan liep, afhankelijk van de rol die je had te vervullen in de

applicatie, ben je verplicht om dagelijks in te loggen en dingen te doen, dus of je daar blij wordt van de applicatie is een andere vraag

P4: Ik kan mij ook voorstellen dat in certain times of the day, dat dat ook uitmaakt. Als je voor het werk gaat met de applicatie bezig bent, is wel een commitment dat je dat uur extra slaap niet pakt zeg maar

P1: Ja je kan ook nog denken: hoeveel meer gebruikers er bij komen per maand

P3: Maar dat heeft meer met growth te maken dan met usability denk ik. Ik snap wat je bedoelt

P1: Ja, want als je een kleine onderneming hebt is het dan gelijk niet usable

P4: Ik denk dat het wel een goede indicator is

P2: Maar met je gebruikers opzich zegt het wel wat, adoptie tijd

P3: Waar ik ook overna denk is de vindbaarheid van bepaalde dingen, logischheid. Lastig misschien

P2: Het aantal clicks tot je bent bij het punt waar je wilt komen

P3: Dus bijvoorbeeld ik heb iemand die ergens zijn naam wilt veranderen, hoeveel clicks doet hij erover ten opzichte van iemand anders, hoeveel clicks doet hij erover. En wat is de meest optimale route? Misschien kan je er binnen twee clicks komen, sommige mensen hebben tien keer geclicked voor ze pas zijn waar ze willen zijn, terwijl ze het ook in twee clicks hadden kunnen doen. Dus de time to goal, zoiets

R: Is het hier de time of het aantal clicks wat?

P3: Misschien beide natuurlijk

P4: Tijd per activiteit, per applicatie activiteit. Denk ik wel dat het ook tijd is, maar ook

P3: Gebruiksvriendelijkheid dan ook: is het logisch wat hier staat. Doe X Y testing

P2: A B

P3: Hoe snel doet deze gebruiker het, hoe snel doet deze gebruiker het. Op die manier kan je usability ook heel goed testen

R: Oke, nog twee minuten voor usability

P1: Het is best lastig om het echt in een getal te krijgen toch

R: Het is goed dat het lastig is, anders had ik geen thesis

P3: Ik denk dat heel veel wat we hier hebben wel onder usability werkt

P4: Ik ben nog steeds een voorstander van brain activity en biometrische gegevens. Als je het voor elkaar krijgt

P3: Je bent een AI fanaat he jij?

P2: Heel eerlijk, ik denk dat usability, dat je daar de meeste metrics op kan pakken die objectief zijn, en daar iets mee kan bedenken

R: Interessant, waarom?

P2: Ik heb het idee dat er meer facetten zijn aan usability, ik bedoel zoals ook in de beschrijving staat, het gaat om efficiency, het gaat om satisfaction. Dus voor elk van die subdoelen kan je. Het volgt het gedrag, en gedrag uit zich in data, en daarom kan je er veel meer metrics over bedenken denk ik

P1: Aan de andere kant is usability inherent ook vaag denk ik. Je kan er metrics van maken volg ik je wel in

P2: Zie je wel het bos door de bomen met alle metrics die je hebt, en volg je dan wel het goede is dan de vraag

P1: Ik bedoel daar zit wel iets subjectiefs in: jij kan iets heel makkelijk vinden in een tool en jij niet, terwijl het dezelfde tool is

P4: Nog 25 minuten jongens

P3: Het heeft ook te maken met hoe vaak je in de applicatie gezeten hebt. Op een bepaald moment weet je wel hoe het werkt. Maar als je dan een nieuwe gebruiker hebt die in je systeem moet kijken, dan is je usability niet hoog. Maar dat is heel lastig te peilen

P1: Hebben we hem van de data as ook gedaan? We zitten bij deze ook wel heel erg op de user interface. Dashboarding. Als je via de data as kijkt, usability? Het aantal datasets dat je moet combineren

P2: Als ik kijk naar data qua usability zit ik vooral te denken aan compatibility. Als die hoog zit qua metrics vind ik het veel makkelijker om te gebruiken

P3: Is het gemodelleerd, vanuit een data hoek toch. Dan kan je toch vrij makkelijk je usability omhoog krijgen omdat je daarin ervaren bent

P1: Ja, en wat is dan de metric in een getal

P3: Een of nul

P1: Ik bedoel, is het gemodelleerd vraagteken, ja of nee dat kan een metric zijn

P3: Als je een heel grote database hebt en veertig procent is gemodelleerd dan heb je wel veertig procent ratio dus dan

P1: Misschien moet zoiets er dan bij maar hoe

P2: Ik weet niet hoe jullie dat willen opschrijven

P3: Ik zit even te denken

P1: Aantal verschillende datasets?

P3: Is het gemodelleerd met huidige standaarden, dat is natuurlijk. Heeft ook heel erg te maken met huidige standaarden, misschien is het over tien jaar weer heel anders, dat zou ook kunnen

P1: Daarom zeg ik aantal verschillende datasets, als we snel iets moeten opschrijven wat binnen de tijd past. Ik bedoel dat het een beetje lastig is om dat erin te verwerken, dus laten we zeggen verschillende datasets. Ik bedoel niet het aantal datasets, het aantal verschillende datasets. Drie dezelfde maakt niet zoveel uit, maar als de ene set Kimble is en de andere is gewoon uit een database en de derde is DataVault, dan denk ik dat je een heel probleem hebt met usability

P3: Licht aan de gebruiker, maar over het algemeen wel ja

R: Alright, laten we doorgaan naar de volgende. Dat is reliability, eerder ook al besproken, de "degree to which a system, product or component performs specified functions, under specified conditions, for a specified period of time"

P4: Hoe vaak iets opgestart wordt achter elkaar, in een bepaald tijdsbestek? Reliability

R: Wat maakt je systeem reliable?

P4: Of het accuraat is. Heb je up to date informatie. Als jij ervanuit gaat dat 's ochtends om negen uur bepaalde datasets klaar zijn, en om negen uur, of tien uur log je in en je kijkt, alle KPI's zijn goed, en om elf uur log je in en je komt erachter dat de KPI's niet goed zijn en je naar de dag ervoor keek

R: Dus wat je eigenlijk zegt: doet de functionaliteit, in dit geval de data, het op een gegeven moment. Dus dit is het gedeelte "performs specified functions, for a specified period of time"

P3: Doet het wat jij verwacht eigenlijk

P2: Onder de bepaalde omstandigheden

P3: En als dat het niet doet, is er een back-up plan voor

P2: Ik denk, de hoeveelheid back-up opties die je hebt, die zorgen ervoor dat reliability hoog is

P3: Dat klopt, als je gewoon duidelijk aangeeft van, pas op, dit is de dataset van dan

R: Is doet het wat verwacht wordt, die is eerder ook al genoemd onder functional suitability, wat zou hier het verschil zijn in wanneer je hem voor functional suitability pakt en wanneer voor reliability?

P2: Error handling denk ik. Dingen werken niet, maar dat je hem wel afvangt als in

P3: Misschien de hoeveelheid logging per functie, dat je het kan monitoren. En het verschil is hier of het reliable is, daar ligt wel een heel groot punt. Je had het over de functionele toch, of die andere?

R: Functional suitability versus reliability

P3: Of iets functional is of reliable daar zit wel een heel groot verschil in, want een fiets met vierkante banden dat fietst niet, maar ik weet wel dat het een fiets is met vierkante banden.

En als je er dan op fietst en alles valt uit elkaar, is het niet reliable. Heeft het niks te maken met die banden uiteindelijk. Dus daar zit wel een cruciaal verschil om het even te vertalen

P1: Als ik kijk even naar opdrachtgever zijn er een hoop opdrachtgevers die kijken heel sterk naar consistentie, naar betrouwbaarheid. Je levert het vorige week op, over oktober 2022, je levert vandaag op, oktober 22, en november 23

P2: Dus consistentie, het verschil tussen toen en nu

P1: Als je hetzelfde nog een keer levert, dat het dan ook hetzelfde resultaat heeft. Maar wat is daar dan de meeteenheid in. Aantal keer

P3: Reliability

P1: Aantal keer in de tijd dat het resultaat overeenstemt. Dat is ook wat ik bij klanten zie. Maar ook echt de vraag: vorige keer zei je blablabla

P3: En dan wordt een dataset gereleased en dan is het getal anders en zeggen ze: huh waarom is dit? Gister heb ik nog een screenshot, gisteren was het wel

P1: Nee precies, het andere betrouwbare is, bij data kun je het hebben over de nauwkeurigheid, ik weet niet of we die kant op willen maar ik kijk naar jullie: het aantal decimalen van een veld, of

P2: Ja, dat sluit aan bij die consistentie toch een beetje?

P1: Als je vraagt, hoeveel auto's heb je geleverd en je zegt: ja, 1.6 miljoen, en morgen vraag ik het je, en je zegt 1.6 miljoen. Dan ben je heel betrouwbaar volgens de vorige definitie.

Maar is het nou 1.6 miljoen en drie auto's, of is het 1.6 miljoen en zeven auto's

P2: Precies, waar ik dan aan zat te denken is, stel je begint met 1.6. Dat zijn je eerste opties die je hebt. En later heb je meer functionaliteit toegevoegd waardoor je veel meer punten achter de komma hebt waardoor het 1.6543890 wordt, dan zit het antwoord misschien, er raakt informatie kwijt. Dus het was misschien precies hetzelfde, maar de consistentie wordt erdoor verstoord

P1: Ik zit alleen zelf te twijfelen, is het reliability, is het nauwkeurigheid of is het net iets anders. Als ik kijk welke dingen we nog hebben, en wat we nog hebben staan. Als ik op het niveau van dit niveau kijk is dit wel degene waar die bijhoort. Dus zet maar neer: aantal decimalen in veld

P2: Opzich wat P4 net zei, het aantal bits per dataveld

P1: Wat hebben we nog niet voor reliability? Ja, of iets heeft hij ons toevallig net een klein beetje iets gestuurd. Er hangt daarboven iets van een ISO standaard. Gaat mij niet om die, maar in het algemeen. Het aantal standaarden waar het product aan voldoet. Als je zes ISO standaarden hebt zal het wel een heel goed product zijn. Dat hoeft dan ook niet, maar het zegt wel iets

R: En wat is dan precies de relatie met reliability?

P1: Objectieve kwaliteit. Aantal keer dat je voldoet aan objectieve kwaliteitsnormen

P2: Maar die hadden we eerder ook al?

P1: Hadden we die al?

P2: Ik weet niet waar. Die hadden we ook al bij compatibility dacht ik, maar ik zie hem er niet tussen staan. Maar toen hadden we gezegd, aantal niet standaarden

P1: Ja oke, maar dat is meer een echt technisch iets dit. Hier hebben we het meer over

R: Reliability is eigenlijk, in welke mate werkt een product nog steeds, wanneer je hem voor een gegeven tijd onder gegeven conditions stopt

P4: En is het ook reliable wanneer het live gaat in productie. Je kan iets maken en het werkt helemaal top, en boom je released naar productie, er gaan duizend mensen mee werken en alle voegen vallen uit elkaar

P2: Maar dan kom je in de buurt van performance efficiency. Dat is stress testing, van te voren doen om reliable te zijn

P3: Ja, dat klopt. Valt onder beide

P2: Ik schrijf hem nog een keertje op, want ik vind hem wel, hij klopt wel

P3: Want hoe kan je op het systeem relyen?

P2: En misschien juist het verschil in stress testing van nu en vroeger, dus altijd gerelateerd aan: we hebben nu even veel users als we bij de test hebben gedaan van drie maanden terug. Maar krijgen we dan ook hetzelfde resultaat. Dus verwachting ten opzichte van performance

P3: We krijgen in een keer andere resultaten, of elke gebruiker krijgt andere resultaten omdat de data connectie die gemaakt moet worden er ergens een fout in zit waardoor bepaalde dingen niet goed door komen bijvoorbeeld, dat zie ik ook wel gebeuren. Dat twee mensen op hetzelfde moment inloggen en andere resultaten krijgen, dan is het niet reliable. Dat kan te maken hebben met zulk soort dingen

P2: Dus het aantal verschillen per units die opgestoken worden

P1: En dit triggerde mij. Kan het ook zijn dat we zeggen? Het aantal gebruikers gewoon. Simpele metriek. Hoe groter het aantal gebruikers, hoe betrouwbaarder het systeem wordt

P2: Nee, juist niet toch

P3: Nee, maar wel als het aantal gebruikers hetzelfde blijft en de output blijft hetzelfde. Je hebt meer mensen erop werken dus dan is het meer reliable, dus ik snap wel

P2: Ik snap niet wat je bedoelt

P1: Gewoon het aantal gebruikers, dus als je een product hebt gemaakt. Een software design tool zelf. Als je met honderd ontwikkelaars eraan werkt of met tien duizend. En goed het is dan wel de aanname een succesvol product, het werkt gewoon

R: Het hele idee wat je voorstelt, als een product honderd gebruikers heeft, en het product heeft duizend gebruikers heeft en het werkt nog steeds, dan is het beter omdat het duizend gebruikers heeft. Volgens mij is dat wat P1 zegt

P1: Is ook wel iets wat we doen, ook in de BI wereld waarin we zitten

P2: Heel effectief is het dan toch gewoon stress testing

P1: Nee, ik vat wel wat anders onder stress testing op. Of het onder extreme condities nog werkt, maar dat is niet het aantal gebruikers. Het kan ook heel veel data zijn met maar een gebruiker. En het kan ook zijn dat de markt verandert

P2: Ja maar hetgene wat je net zegt denk ik dat het aantal gebruikers niet voldoende is, want je hebt intensieve gebruikers en niet intensieve gebruikers

P1: Maar ik zeg ook, het is niet genoeg. Het is een metriek gewoon. In feite gewoon, als we alle metrieke naast elkaar zetten, zegt dat iets over de reliability, en misschien niet alles. Sterker nog: we weten dat het niet over alles wat zegt, want we hebben beperkte tijd maar

R: Oke, in verband met de tijd, zullen we doorgaan met de volgende. Portability, portability is de “degree of effectiveness and efficiency to which a system or product can be transferred from one usage environment to another”

P2: Dus er zijn ook specifieke tooling natuurlijk die dat makkelijk maken. Dat het juist bruikbaar wordt op elk systeem \*inaudible\*. Maar bijvoorbeeld dat als je een applicatie ontwikkelt, en het beschikbaar stelt op een browser, dan hoef je niet onderliggend na te denken werkt het op Android, werkt het op

P3: Dat moest wel hoor, maar tegenwoordig werkt het beter

P2: Ja inderdaad, maar mobiele pagina's meer van. Het komt wel beschikbaar

P3: Elke browser heeft zijn eigen manier van het handelen van bepaalde codes destijds

P2: Ja fair, maar je bent wel los van een operating system. Je bent wel meer verbonden aan het type browser dat je gebruikt en vanuit daar minder portable op dat vlak misschien

P3: Lastig, deze vind ik lastig

P1: Ik had er wel een paar. Aantal aanpassingen die de ontwikkelaar moet doen voordat die het om kan zetten naar een andere omgeving. Als het alleen een druk op de knop is en hij compileert het naar een andere omgeving dan is het programma heel portable, want hij kan het direct omzetten

P3: Dan heb je het over bijvoorbeeld een dev naar test, test naar acceptatie, of productie, wat dan ook

P1: Kan ook, dat is er ook eentje, dus aantal handelingen, aantal benodigde handelingen door de ontwikkelaar. En ik dacht ok aan de tijd. De tijd die het duurt om het over te zetten. Dus het aantal handelingen kan het zijn. Het hoeven niet perse heel veel handelingen te zijn. Maar het duurt heel lang om te compileren, kans op fouten, en heel extreem geval. Stel dat het met nul handelingen kan. Maar om het om te zetten moet de computer drie maanden rekenen. Vind je het dan nog hele portable software? Het is gelukt maar er is een verandering morgen, moet ik weer drie maanden wachten. Dus tijd die het duurt om het over te zetten. Dat maakt ook wel uit

P3: Dat is wel een goeie. Wel een extreem voorbeeld maar ik snap wel wat je bedoelt ermee

P1: Ja daar kwam ik toevallig op bij een van die anderen, daar had ik even niet zo snel een beeld

P2: Ergens is de modulariteit hier ook een ding. Alleen het ene model te hoeven vervangen om de rest van het systeem bruikbaar te houden. Stel je hebt een open data connector. De ene keer komt het

P3: Ik snap wel wat je bedoelt. Een beetje wat ik met die CO2 monitoren heb. Als je op een plek wat wilt veranderen moet je het op 140 verschillende plekken iets veranderen om het goed krijgen in plaats van dat je ergens een parametertje verandert

P1: En dat is dan nog anders dan het aantal handelingen die een ontwikkelaar moet doen, maar het is het aantal handelingen?

P3: Het is het aantal bewerkingen

P1: Aantal bewerkingen per wijziging

P3: Dus als je een wijziging moet doen door een feature, dus bijvoorbeeld, haal de data nu bij deze bron op. Of je op een plek alleen de databron naam moet aanpassen, of dat je op heel veel plekken in de codebase

P1: Ja, dan is de code die jij of een voorganger gemaakt had niet heel portable als je het 140 keer moet doen, ookal is het maar een ding die je moet veranderen. Dus die hebben we ook zeker nodig. Kijkend naar de portabiliteit. Het gaat om een nieuwe omgeving, aantal omgevingen die de software ondersteunt. Dat gaat dan wel uit van software die, nieuwe software is dan automatisch in het nadeel, dat is natuurlijk niet perse zo



P2: Afhankelijk van wat je dus onderliggend gebruikt

P1: Dus eigenlijk is het wel een goeie, aantal omgevingen die de software ondersteunt, of kan ondersteunen, dat is wat vager. Aantal omgevingen die het ondersteunt, dat is toch portable? Het aantal omgevingen die de software ondersteunt, of waar de software op kan draaien, misschien is dat beter

P2: Ja

P3: Ja

P1: Talen, is dat ook nog iets waar we naar moeten kijken, of is dat een ander soort portabiliteit om zo te zeggen?

P2: Talen, valt dat ook hieronder?

R: Talen, als in?

P1: Een nederlands, engels, duits, dus ik bedoelde gebruikers talen maar je kan het ook hebben over, programmeertalen, ja dat had ik nog niet bedacht

R: Ik zou zelf zeggen van niet, als we puur kijken naar de definitie. Compatibility gaat meer over exchange of information. Dit gaat meer over, kan je een product van een omgeving naar een andere veranderen. Dus ik zou talen er niet bij zetten. Als jullie een goede reden hebben om het wel te doen is het natuurlijk wel

P4: Hoe ga ik dit zeggen. Hoe je een hele nieuwe taal moet leren om het voor elkaar te krijgen, zeg Docker, of dat je vooral een user interface hebt waarin dat voor je gedaan wordt \*inaudible\* of dat je er drie jaar mee bezig bent

P2: Dat je het in ChatGPT invult en zegt van schrijf maar een functie dat het werkt in een ander ding terwijl je helemaal geen ervaring hebt maar poep code die je erin kan plakken

P1: Maar dat zit dan toch ook al in het tijd van het overzetten?

P2: Maar dit gaat over de domeinkennis van de developer

P4: Ja, dit gaat over als je het van de ene naar de andere omgeving moet overzetten en om dat voor elkaar te krijgen moet je dat in een specifieke taal doen \*inaudible\*. Dan is het een stuk moeilijker dan als je een interface hebt, een GUI, waarin je kan slepen bij wijze van spreken

P2: Dus de moeilijkheidsgraad

P4: Van de transitie ofzo

P1: Maar wat is dat voor metric?

P4: Wat je zegt klopt hoor, het zit ook wel in de tijd, alleen de tijd kan ook relatie hebben met iets anders dan de context die je meet

P1: Nee klopt, maar hoe meet je de complexiteit die je noemt iets concreter?

P4: Ja

P1: De tijd die je nodig hebt per verandering ofzo?

P4: Of de kwaliteit. Dus hoeveel kwaliteit die je nodig hebt om iets voor elkaar te krijgen

P3: De mate van documentatie

P1: Aantal tussenstappen die je moet doen

R: Laten we nog een minuut over portability doen, want we zitten bijna aan de tijd

P3: Ik moet op dit moment as we speak een systeem migreren van SAS naar Azure en dat is echt een drama soms, omdat er gewoon helemaal geen documentatie is

P4: Vaardigheden die nodig zijn misschien

P1: Number of skills

P4: Aantal vaardigheden

P1: Dan houden we hem iets breder, maar dat dekt wel af wat jij net noemde

P4: Laatste?

R: Laten we dat doen. Dan kunnen jullie ook de werkweek afsluiten. Oke, de laatste is security. En security is de "degree to which a product or system protects information and data, so that persons or other products have the degree of data access appropriate to their level of authorization"

P2: Ja. Aantal lines of defense die ter beschikking zijn?

P4: Prima

P3: En misschien ook de diversificatie van verdediging. Dus bijvoorbeeld verschillende soorten

P2: Zoiets als dat je een fysieke device moet hebben, een sleutel. Een bioprint

P4: Multi factor authentication bijvoorbeeld. Aantal soorten. Diversificatie van

P1: Ik schrijf ze eerst wel even op

P4: En dan misschien fysieke combineren

P2: Ja, want je wilt dus, je wilt het tijdgebonden hebben, persoonsgebonden hebben, en objectgebonden hebben

P3: Row level security?

P2: Hoe heet dat?

P1: Ja, aantal soorten beveiliging, maatregel?

P2: Aantal soorten beveiligingen die locatie, persoon en weet ik wel niet wat allemaal uit elkaar haalt

P1: Ja, maar dat valt er allemaal onder

P2: Oke, aantal soorten beveiligingen

P1: Daarbij is dan de aanname hoe meer soorten hoe beter beveiligd, dat hoeft niet. Ze kunnen elkaar ook tegenwerken. Maar ik bedoel maakt niet uit, het zegt wel iets over, maar misschien niet alles, daar hebben we andere dingen voor. Ik denk ook het aantal datalekken per jaar, als we het heel concreet maken. Dan gaat het er wel om. Het moet hebben zijn plaatsgevonden, ook dat zegt niet alles. Dat het nog niet is plaatsgevonden betekent niet perse. Maar in ieder geval wel een goeie

P3: Relatief tot het aantal aanvallen

P1: Ja, weten we dat ook

P3: Ja kijk als je twee aanvallen hebt en je hebt geen datalek ten opzichte van iemand die een miljoen aanvallen heeft en twee datalekken dan basically is het net alsof diegene het heel slecht doet

P1: Ja precies, dan combineer je ze een beetje

P2: Klinkt goed

P1: Ik bedoel aantal datalekken kun je, dat weet je wel. Maar aantal aanvallen ja

P2: Misschien aantal points of failure? Als in hoe dependent het is aan andere dingen

P4: Ja, zo, dat is ook wel een goeie, zeker. Anonimisering van de data misschien überhaupt?

P1: Dat zit niet bij security, dat zit meer bij privacy

P4: Nee, dat is waar

P1: Ik zat te denken omgekeerd, hoe minder tijd, als een applicatie maar twee uur per dag beschikbaar is versus 24 uur, dan kan je zeggen

P3: Gewoon downtime dan. Dus als die meer is dan kan je ook niet \*inaudible\*

R: Hoort downtime bij de veiligheid of bij de reliability van een systeem?

P1: Dan ook nog een keer dat het reliability is en bovendien komen we op een systeem dat heel veilig is omdat het niet aan het internet zit. Dat soort onzin komen we dan mee

P4: Wij hebben alles on premise dus wij zijn goed bezig

P1: Precies ja

P4: Ik ben wel een beetje leeg

P2: Maar jullie zeiden nog een dingetje dat ik niet kon opschrijven omdat ik geen post it note had

P1: Het aantal soorten beveiligingsmaatregelen, en we zeiden aantal beveiligings

P2: Ik weet het niet, jullie hadden het erover. Ik heb het letterlijk niet gehoord

R: Het staat op de opname, dus we kunnen hem ook nog achteraf toevoegen. Dit waren ze.

Dan wil ik jullie in ieder geval heel erg bedanken voor het mee doen aan mijn onderzoek.

Echt heel erg gewaardeerd. Dan ga ik nu de opname stoppen.