

R: Oke, dan heb ik nu de opname gestart. Allereerst bedankt voor het deelnemen aan deze sessie. De sessie zal de volgende agenda hebben. Allereerst zal ik uitleggen wat dit onderzoek doen, wat we gaan doen aan deze focus group, wat voornamelijk het bepalen van kwantitatieve measures is. Het onderzoek richt zich erop om kwantitatieve waarden te vinden voor de ISO 25010 standaard. De standaard is een hiërarchisch model dat bestaat uit drie verschillende niveaus. Op het eerste niveau hebben we software product kwaliteit. Op het tweede niveau hebben we kwaliteitskarakteristieken, op het derde niveau zijn subkarakteristieken, waar we ons tijdens deze focus group niet mee bezig gaan houden. Dus de taak die we zometeen gaan doen is het vinden van kwantitatieve en objectieve measures voor software product kwaliteit. Ik benadruk dus dat het belangrijk is om objectieve measures te vinden, er is gerelateerd onderzoek dat zich heeft gefocust om andere measures te vinden, maar die zijn niet objectief. Dus een voorbeeld van een objectieve measure, als je bijvoorbeeld kijkt naar de kwaliteit van een auto, hoelang gaat een motor mee, dat is een objectieve measure, terwijl een niet objectieve measure zou bijvoorbeeld zijn: hoe fijn vind je dat de auto rijdt? Is dat een duidelijk voorbeeld? Snappen jullie het verschil tussen objectieve en niet objectieve measures?

P1: Ja

R: Oke top, dan denk ik dat het nu tijd is om ons te gaan focussen op de discussie over de acht kwaliteitskarakteristieken. Dus dan beginnen we met functional suitability. Wat ik ga doen, voor elk ga ik jullie een definitie voorlezen, en dan is er ruimte voor discussie. We hebben in ieder geval post-its en een pen, waarop we mogelijke suggesties kunnen opschrijven. Oke, de allereerste kwaliteitskarakteristiek is functional suitability, ik lees de definitie in het engels voor omdat die het meest overeenkomt met hoe het in de standaard beschreven is. Functional suitability is de "degree to which a product or system provides functions that meet stated and implied needs". Oke? Dat is de definitie. Als jullie nog vragen hebben over de definitie of er meer uitleg nodig is, laat het ook weten. Dus dan nu de eerste vraag. Hoe zou je functional suitability mogelijk kunnen meten?

P2: Hoe je het meet?

P3: Nogmaals even lezen.

P2: Wat de definitie volgens mij al zegt zijn de stated and implied needs, wat betekent dat er al gedefinieerd is hoe het, wanneer het functioneert, wanneer het goed functioneert. Dus dat betekent dat een van de dingen om het te kunnen meten moet zijn de definitie van wat wil je ermee bereiken. Je moet dat al kunnen kwantificeren om te kwantificeren in hoeverre het product wat je levert voldoet wat je ervan gevraagd hebt

P3: Wat is de functionaliteit, wat moet het kunnen

P2: Ja. Dus zeg je maakt een website om iets, je kan iets bestellen, je hebt een webshop. In hoeverre kan je doen bij die webshop wat je wilt bereiken

P1: Maar hoe ga je dat dan ook objectief meten, dat is een beetje het pijnpunt

P3: Volgens mij is dat heel recht toe recht aan toch. Je hebt een doel: je wilt iets bestellen. En achteraf kan je dan meten, kan je iets bestellen ja nee

P2: Gele ballonnen punt nl. Kan ik gele ballonnen

P3: Kan ik gele ballonnen bestellen

P2: De vraag is dan een beetje ook, in hoeverre kan ik hoofd en bijzaak van elkaar onderscheiden, dus stel je kan gele ballonnen bestellen, oke prima. Maar vervolgens liggen wel al mijn gegevens op straat. Dat lijkt mij in principe bijzaak van wat je functioneel wilt bereiken. Maar het lijkt mij wel belangrijk

P1: Dat zit meer in de security hoek zou je dan zeggen

P2: Maar zo zijn er misschien wel meer dingen

P1: Maar ik zit dan te denken van, stel je kan een ballon bestellen, maar een op de twee keer dat je op de knop klikt gebeurt er niks. Is dat dan een puntje onder reliability? Of is je functionaliteit niet goed

P2: Afhankelijk van wat je beschreven hebt, stel je hebt hem zo gebouwd dat je opgegeven hebt. Ik wil dat die past wat doet bij de derde keer. Dan doet die wat je beschreven hebt wat die zou moeten doen, lijkt mij niet heel nuttig, maar ik denk dat het daar onder valt toch

P3: Als je doel SMART genoeg omschreven is, dat die altijd, bij elke klik een gele ballon bestelt, dan zou je dat heel objectief kunnen afvangen, en meten

P2: Dan heb je nu wel een heel concreet voorbeeld te pakken wat makkelijk te definiëren is. Maar in principe is dat wel

P3: Die zou je ook prima op kunnen knippen in sub-functionals. Een beetje zoals wij in ons huidige project unit tests maken, zou je dat ook voor een stuk software hetzelfde. Hij moet iets doen. Ik test of hij iets doet, ja of nee. Volgens mij moet je hem zo simpel maken

P2: Jij zegt wel een heel concreet voorbeeld, ik ben wel op zoek naar een niet concreet voorbeeld, die zijn er ongetwijfeld. Als je bijvoorbeeld kijkt naar, wel een duidelijk voorbeeld maar iets vager, een dashboard dat ik aan het bouwen ben

P3: Die moet meerdere dingen kunnen, maar dat zie ik dan als een sub-functional zeg maar. Dus hij moet en in dit geval, en een kaart kunnen laten zien, en kunnen filteren op een datum. Als je dat gaat omvatten in een ding wordt het heel abstract. Maar volgens mij moet je juist willen moduleren, en ieder sub-dingetje kan je dan weer toetsen

P2: Dus als je er een objectieve meetwaarde aan wilt hangen zeg je dan: er zijn acht dingen die die moet kunnen doen. Zes van de acht doen het, dus daarmee heeft hij driekwart score behaald

P3: Bijvoorbeeld

P2: En stel je voegt daar een negende functionaliteit aan toe die je niet gedefinieerd hebt, maar hij kan het wel, maakt het het product ook slechter ja of nee. Hij doet in principe wat je verwacht dat die doet. Maar ik kan mij ook wel voorstellen, we hebben bepaalde interfaces mee gewerkt die alles konden, dat is niet heel gebruikersvriendelijk. Dat zit misschien meer in de gebruikersvriendelijkheid hoek. Maar dat schuurt wel met in hoeverre je functioneel je doel kunt bereiken

P3: Dan is ook de discussie, waarom heb je dat. Kijk als je iets gaat maken, als je iets koopt en het zit erin kan je het op de koop toenemen. Waarom zou je een negende functionaliteit toevoegen als je het niet nodig hebt. Als het niet je doel dient

P2: In hoeverre een product doet wat gedefinieerd is, is er ook een categorie die meer op de usability ingaat?

R: Jullie kunnen hier kijken naar alle verschillende. We hebben dus functional suitability, portability, dus in hoeverre is het draagbaar tussen systemen. We hebben reliability, dus in hoeverre is het bijvoorbeeld fault tolerant. We hebben maintainability, hoe makkelijk is het om je software te onderhouden. We hebben security, dus in hoeverre is de access tot data goed. We hebben usability er tussen zitten, hadden we het net al over. Compatibility, dus in hoeverre kan het gedeployed worden, is het interoperable met elkaar. En performance efficiency, hoe goed performed jouw product onder gegeven resources

P2: Dus eigenlijk heel interessant, want functional suitability en usability schuren daar wel, niet perse, maar die raken elkaar heel erg aan. In de zin van. In hoeverre kan je je hoofddoel bereiken met je product. Die zou je meer verwachten in de functional suitability hoek. Terwijl, welke middelen heb je tot je beschikking omdat product makkelijk te gebruiken

R: Het kan ook zeggen dat jullie zeggen, deze twee zijn zo dependent op elkaar dat we een concrete metric hebben voor meerdere kwaliteitskarakteristieken, dat moeten we dan even

aangeven. Dat zou natuurlijk ook een antwoord kunnen zijn als jullie zeggen: het is niet mogelijk om ze los te trekken van elkaar want er is een afhankelijkheid

P3: Naar mijn mening niet, als je heel zwart wit bereikt. Kan je functional suitability heel zwart wit bekijken. Je kan het of je kan het niet. Maar hoe makkelijk kan ik het of kan ik niet, daar heb je wel zeker een relatie

P2: Als ik het goed vertaal. Een is kan ik mijn hoofddoel bereiken, kan het wat ik verwacht, doet het wat het zou moeten doen, wat ik ermee wil bereiken, zal ik zo zeggen. En in hoeverre gaat dat eenvoudig, gebruikersvriendelijk

P3: Daar gaat een ander sausje overheen zeg maar, zo zie ik het, usability

P2: Een andere as, ja. Is er dan misschien waarde in het hangen van verschillende gewichten aan verschillende functionaliteiten. Zeg maar er zijn hoofd doelen en subdoelen

P3: Ja zeker

P2: Want als je een concreet getal er over wilt houden aan

R: Wat uiteindelijk dit onderzoek ook gaat doen is dus een heel gewichtenmodel hieraan hangen. Dus vanaf jullie perspectief zou je die dan allemaal kunnen losrekken. Als je zou stellen dat er verschillen zouden zijn in het belang van de feature. Dan zou je het ook los kunnen trekken in meerdere metrics

P2: Dus om het gele ballonnen verhaal er weer even bij te pakken: je zou kunnen zeggen. Ik kan enerzijds gele ballonnen bestellen, en anderzijds een retour formulier kunnen invullen om gele ballonnen terug te sturen, uitprinten noem het maar op. Een functionaliteit die ook belangrijk is, maar minder vaak voorkomt dat iemand gaat bestellen, dus is het gewicht wat lager. Dus ik kan mij voorstellen dat je daar dan minder gewicht aan gaat hangen

R: Dus hoe zouden we dit kunnen omschrijven als concrete metrics? Dus over het algemeen het idee van een objectieve metric is dat je er dus een numerieke waarde aan kan hangen. En het belang van een numerieke waarde is, je hebt waardes die relatief ten opzichte van iets anders zijn. Je hebt ook waardes die relatief aan een standaard kunnen zijn. Denk weer bijvoorbeeld aan het voorbeeld van een auto, kan natuurlijk: bij wijze hoe snel kan je auto optrekken ten opzichte van bijvoorbeeld het gemiddelde van een personenwagen

P1: Direct om met een concreet algoritme aan te komen is ook weer teveel gevraagd, maar je kunt het op zijn minst het aantal functionaliteiten wat geleverd is, versus het aantal dat gevraagd was. Is het meer of minder, dat zal zeker impact hebben. Wat vraag ik, en in hoeverre voldoe ik daaraan. Niet platte aantallen. Ik vraag om A, ik krijg A tot en met Z. Dat is leuk en aardig, maar ik heb nog steeds niet gekregen wat ik wil krijgen

P2: Ik denk wel interessant

R: Dat zou een aparte metric mogelijk kunnen zijn

P2: Die vind ik wel interessant, want kan ik nog steeds doen wat ik met het product doen wat ik wil doen is de vraag. Stel je wilt bestellen maar er is een aantal tooltje bij die je vertelt wat de bezorgtijd is, of hoe groot de voorraad is. Het helpt je niet misschien bij het bestelproces, maar wel usability. Puur functioneel is hetzelfde, maar hoeveel je er in steekt om het te maken is wel interessant. Je vraagt om een ding en je krijgt honderd dingen, dan is het veel te duur geweest

P3: Dan kom je wel bij het stukje 'meets stated and implied needs', als de implied need er niet was, ik hoef niet te kunnen zien hoelang het duurt tot die er is, de gele ballon. Dan is het overkill, dus nee, dan voldoet het niet aan de implied needs

P2: Wat als je hem meer bekijkt vanuit een overkoepelend doel, dus denk daarbij aan. Je hebt een hoofddoel waaraan je wilt voldoen, en je hebt functionaliteiten die daaronder hangen, subdoelen zou ik zeggen. Zijn er dingen die potentieel kunnen bijdragen aan het

hoofddoel, ook al hebben we die niet specifiek gedefinieerd als functionaliteit. Dus maakt het proces wat je er mee wilt bereiken makkelijker

P1: Je wilt kunnen meten in hoeverre de functionaliteit die je hebt gemeten bijdraagt aan het hoofddoel zeg maar

P3: Is je hoofddoel dan misschien niet goed?

P2: Ik ken iemand die heeft bij Marktplaats gewerkt en hij zei dat mensen die bij Marktplaats werkten mochten bijdragen aan het zo snel mogelijk online krijgen van een advertentie, ongeacht wat het is. Als iemand een idee had, klopt het idee dan gaat we het doorvoeren heel plat gezegd. Je hoofddoel is, en dat is ook wel gecompliceerd, die advertentie online krijgen. Maar wat die functionaliteiten zijn dat zijn hele kleine dingetjes. Het zijn dingen in het proces die sneller zijn. Dus ergens heb je wel een hoofddoel wat je wilt bereiken, en wat je daarvoor gebouwd hebt, bijdraagt aan het hoofddoel. Maar je kan enerzijds zeggen, ik ga al mijn functionaliteiten definiëren, of in hoeverre voldoen alle dingen die ik heb gebouwd aan het bijdragen van het hoofddoel

P1: Dat hangt er heel erg vanaf of je doel een proces of product is. Daartussen maak je een heel groot onderscheid. Want om een proces te kwantificeren heb je compleet andere dingen nodig dan om een product te kwantificeren

P2: Ja, eens

P1: Kunnen we dat op een manier samenvatten?

P3: Het valt en staat allemaal bij, ik wil iets, en ik krijg iets, uiteindelijk. En hoe ga je dat nou objectief meten? Ten opzichte van iets

P2: Dus wat we net zeiden, je kan zeggen ik heb wat dingen waar ik om vraag, wat functionaliteiten waar ik om vraag binnen zo'n product. In hoeverre heb ik het gekregen. Ik heb dingen gekregen, ik heb ook een vierde ding gekregen, waar ik niet om heb gevraagd. Haalt dat dat cijfer dan omlaag, ja of nee, die is interessant

R: Zou het ook zinvol zijn om die samen naast elkaar te meten?

P2: ik denk het wel. Wat jij zei P1, je kan ook teveel krijgen. Het kan ten koste gaan van de usability. Je krijgt nog steeds wat je wilt, maar ergens voldoet het niet aan wat je gevraagd hebt. Dus de stated needs

P3: Is het niet gewoon usability?

P2: Het is ook wel performance efficiency als ik zo kijk naar die laatste

R: We kunnen er natuurlijk ook altijd een opschrijven, en die later moven. Nu we de discussie over usability hebben komen we erachter dat die misschien beter bij usability zou passen dan bij functional suitability, dat is ook mogelijk. Dus het is niet definitief. We kunnen ze opschrijven, en na afloop als we meer discussie hebben ze altijd schrappen. Dat is oke, het is niet wat we in ons model willen hebben

P2: Dus samenvattend wat we net hadden was, in hoeverre heeft men opgeleverd wat men gevraagd is

R: Ja dat is dan dus deze, het aantal geleverde functionaliteiten gedeeld door het aantal gevraagde functionaliteiten als metric

P3: En als alles is geleverd heb je honderd procent

P2: Ja, ik heb om A,B,C gevraagd en ik heb A,B,C gekregen, en ik heb A,B,D gekregen, dus ik heb er nog steeds drie. Dat kan misschien ook wel het ding zijn. Er kan een verschil zijn in: je mist c maar je krijgt d, is dat het meer waard of minder waard. Welke van de twee wegen zwaarder. Dus wil je een extra stap van, missen er functionaliteiten versus overgeleverd maar niet super belangrijk is

P3: Goeie

P2: Stel je definieert een bepaalde functionaliteit, en je wilt er X mee kunnen, en je krijgt iets wat dan kan. Dan is het heel makkelijk, dat ligt eraan wat je gevraagd hebt. Maar stel je krijgt wel wat het zou moeten kunnen, maar de oplossingsrichting is anders dan verwacht. Net zoals dat wij acceptatie criteria opschrijven, vanuit functioneel perspectief moeten gaan kijken. Kan het wat het wil, ongeacht hoe ik het op ga lossen. Dan zit je meer in de technische hoek: performance efficiency, usability misschien. Hoe je het dan opgelost hebt, dat het daar misschien meer schuurt. Dus ik denk wel als je teveel gekregen hebt dat dat effect kan hebben op je functioneel suitability

P3: Dan moeten we die ook toevoegen

R: Zullen we kijken of we er een paar hiervan kunnen noteren, en dan in verband met tijd purposes ook kunnen gaan naar de volgende. We kunnen hem ook aan het einde schrappen. Deze is toch niet zo zinvol als we dachten, voor reden X. Dus degene die we dan zouden kunnen opschrijven, er zijn verschillende dingen gezegd. De twee grote topics die we hadden besproken, dus overdelivery hebben we besproken, en ook het aantal functionaliteiten dat bijdraagt aan je hoofddoel, dat is ook iets wat we besproken hebben

P2: Ja inderdaad, dat is samenvattend wat we besproken hebben

R: We kunnen ze altijd nog schrappen op het einde

P3: Dus we hebben: hadden we gekregen wat we willen, in totaal? En hebben we mogelijk meer, dat is ook een metric toch?

P2: Je wilt echt een nummertje eraan krijgen. Dus hoeveel van de functionaliteiten dragen echt bij aan het hoofddoel, dat is wel een belangrijke

R: Dan zou je zeggen: nummer functionaliteiten gedeeld door nummer functionaliteiten aan hoofddoel. Hoe zou je dat dan kunnen bepalen? Hoe draagt iets bij aan een hoofddoel?

P3: Dan wordt het weer subjectief

P2: \*inaudible\* als je weet van een functionaliteit, in het geval van gele ballonnen punt nl ik weet van je kan gegevens op de site wijzigen. Dat is leuk dat dat kan maar

P3: Ik wil gele ballonnen bestellen

P2: Dat is het doel. Het lijkt mij in sommige gevallen belangrijk, het kan een goede metric zijn, dat je scherpe focus houdt op je hoofddoel. Dus dat je niet een hele suite aan dingen hebt, en dat je dat kan kwantificeren denk ik. Dat je onderscheid maakt tussen visie en missie, enerzijds ik wil nu de beste ergens in zijn. En anderzijds ik wil in 25 25 zoveel procent nieuwe klanten ik zeg maar wat. Iets waarvan je kan zeggen, dit is iets kwantificeerbaars, en ik wil het ophangen aan dat abstracte hoofddoel. Dus je doelen moeten meetbaar zijn, je meetlat is duidelijk en kwantificeerbaar. En de dingen die je net opnoemde wil je daar tegenaan meten

P3: Ik zit even te denken, als ik nu een doel op stel, ik neem Leapsome als voorbeeld, daar kan ik subdoelen aan ophangen. En de mate waarin ik mijn hoofddoel haal is afhankelijk aan of ik mijn subdoelen haal. Zo berekend die dat automatisch. Dus stel je zegt gele ballonnen punt nl, je moet ballonnen kunnen bestellen, je moet gegevens kunnen bijhouden, en hij moet retour kunnen sturen. Als al die drie dingen kunnen. Dat is dus een ja of een nee. Ja het kan of nee het kan niet. Die hebben allemaal dezelfde weging, dat zou je dan kunnen bepalen. Hoe zwaar de ene weegt ten opzichte van de andere. Dat alles telt op naar een functional score. Een score voor je functional suitability. Het kan alle drie: honderd procent score. Het kan twee van de drie: dan zestig procent score. Een van de drie: dan dertig procent score

P2: Het hoofddoel is niet perse op zichzelf meetbaar, maar je hangt subdoelen aan het hoofddoel die wel meetbaar zijn

P3: Zoiets. Dus ga kijken per sub-functional. Ga kijken kan het ja of nee. Dus bijdragen aan je score voor je suitability. Het is wel lastig om dat op te schrijven. Dus je moet weten hoeveel doelen je hebt, hoeveel subdoelen je hebt. Hoeveel functionaliteiten je om gevraagd hebt, en welke

P2: Je kan meten in hoeverre kan ik dat, en afhankelijk van gewicht dat je aan die functionaliteiten gaat hangen kan je de score daar aan hangen. En waar we het net over hadden. Als ik meer heb gekregen dan dat ik besteld hebben, kan dat ook een negatieve impact hebben. Maar ik kan mij voorstellen dat als je rode ballonnen kan bestellen. Je hebt er helemaal niet om gevraagd en het is helemaal niet je business model, dat het negatieve effecten heeft op waar je om gevraagd hebt

P3: Dus aantal functionals gevraagd, en gespecificeerd. Dan wil je dus weten, die elk een bepaalde weging hebben. Ja hoe ga je hier nou een formule van maken zeg maar

P2: Maar is het doel van nu?

P3: Weet ik niet

R: Laten we hem gewoon breed opschrijven. We hebben ook de discussie gevoerd, het kan altijd later

P3: Elke sub-functional is onderdeel van je totale score

R: Oke top, laten we inderdaad met time constraints kijken naar de volgende. De volgende is portability. Portability is in het model gedefinieerd als "de degree of effectiveness en efficiency with which a system or product can be transferred from one usage environment to another"

P3: Dus hoe goed en efficiënt kunnen we een systeem verplaatsen van het ene platform naar een andere. En hoe meet je dat?

P2: En de achterliggende vraag, is er een definitie waarom het in ISO 25010 opgenomen is, waarom het relevant is. Ik kan mij voorstellen dat het per product verschilt of het belangrijk is

R: Dat klopt inderdaad. Het is een algemene guideline, dus voor een specifiek software product kan het belangrijk zijn dat die portable zijn. Het kan ook zijn dat het voor een bepaald product totaal niet belangrijk is. En dan heb je het tweede deel van het onderzoek waarin we er gewichten aan gaan hangen. Dat gaat zich hierop focussen. Dus dit is gemaakt op de aanname dat portability belangrijk is voor een gegeven product

P2: Ik kan mij wel voorstellen dat als het wel belangrijk is dat er nog steeds een scope is waarin het belangrijk is. Dus stel je hebt, ik ga niet zeggen gele ballonnen. Stel je zit in de game wereld en je maakt een game die op het ene platform relevant is, ik wil hem porten naar een ander platform. Maar als je zegt ik doe alleen Xbox en Playstation, want ik vind het niet feasible om het op een Gameboy color uit te brengen want het weegt niet op tegen de resources. Dan is je meetlat ook gebaseerd op waar je scope wel op gebaseerd is. Dus ook hier is je meetlat heel belangrijk.

P3: Dus wat is je meetlat? Dat moeten we meten

P2: Als eerste wil je überhaupt het aantal portable environments waar je naar toe wilt gaan

R: Dat is een beetje de baseline, en daarna kunnen we kijken wat gaan we meten om te voldoen aan die baseline

P2: Ja. Die is interessant. Dan ga je namelijk meten in hoeverre. Als ik denk aan portability denk ik in hoeverre doet mijn product functional en non-functional het nog ten opzichte van waar ik vandaan kom. Dus stel ik ga van platform A naar platform B doet die wat het moet doen

R: Dat gaat dan over effectiveness?

P2: Ja, en ik kan mij voorstellen dat je ook bepaalde non-functionals hebt waarin je wilt zeggen ik wil nogsteeds dat die een bepaalde mate van security heeft, ongeacht of die het

functioneel wel of niet doet, bijvoorbeeld dat je alleen maar gebruik kan maken van het product als je bent ingelogd. Dus ik kan mij voorstellen dat portability op zichzelf ook een factor is waarin je meerdere van deze kaartjes kan meenemen

P3: Reliability is ook belangrijk want hij moet het nog wel doen daarna. Security is belangrijk

P1: Misschien zit ik te simpel na te denken. Maar we hebben een set aan functionals.

Hoeveel functionals doen het nog op mijn geporteerde platform. Stel dat ik er tachtig wilde laten werken en er doen er nog zestig het. Ik denk dat dat zeker belangrijk is. Dus dat we die in ieder geval opschrijven

P2: Nouja, ik kan mij wel voorstellen dat je denkt. Een bepaald platform wel of niet te doen

P1: Tuurlijk. Apple Pay of Google supporten

P2: Dan is de vraag, is dat je doel. Want als je doel is dat je Apple Pay gebruikt of wilt kunnen bepalen

P1: Precies, precies, dat is wel een goeie

P2: Dus enerzijds de scope van portability heeft te maken met nog steeds welke functionaliteiten wil ik ondersteunen. Dus nog steeds heb je de definitie van functional effectiveness

P3: Oke dus het aantal functionals wat werkt op platform A gedeeld door het aantal functionals dat werkt nadat je het overgezet hebt op platform B

P2: Ervan uitgaande dat A je originele target was

P1: In ieder geval, je pakt dezelfde meetlat erbij

R: Ik denk dat hier ook een hele interessante vraag is, die ook tijdens de vorige focus group ter sprake kwam en nu ook kunt stellen is. Over het algemeen als we naar portability kijken kijken we in retrospectieve, dus we hebben iets portable gemaakt. Bijvoorbeeld deze: in hoeverre werken functionaliteiten in een volgend platform. Zou portability voor het transferren naar een nieuwe omgeving ook belangrijk kunnen zijn? Hoe zou je het dan kunnen meten? Wat zou je dan überhaupt meten?

P1: Ligt eraan wat voor soort portability je in gedachten hebt. Als je het hebt over software environments dan zou ik zeggen hou rekening met zoveel mogelijk open source producten zodat je een zo groot mogelijke kast van ondersteuning hebt. Maar als je zegt ik wil dat mijn website gelezen kan worden door een braille apparaat bijvoorbeeld, zou ik rekening ermee moeten houden bij het bouwen van een website: ik wil overal tags aanbieden zodat het automatisch uitgelezen wordt

P3: Maar dat is vooraf lastig te meten?

P2: En dat is voornamelijk technische implementatie niveau. Niet perse functioneel niveau. Hoe definieer ik wat portability is van te voren. Je moet nog steeds weten waar je heen wilt porten. Dus stel ik maak iets van een data pipeline of een cloud platform en ik zeg ik wil dat het zowel op cloud platform A als B is te draaien. Dan hou ik er niet rekening mee dat iemand ooit zo'n data pipeline moet kunnen lezen met een braille apparaat. Terwijl achteraf denk je, dat is wel handig, want ik heb een hele club met data engineers die allemaal blind zijn ofzo ik zeg maar wat. Dat weet je van te voren niet

P3: Maar hoe ga je dat van te voren meten, hoe ga je daar een metric aan de voorkant aan hangen?

P1: Je kan gebruik maken van cross platform tools, zoals Terraform

P2: Dan heb je de scope in gedachten. Er is een reële kans dat het op meerdere cloud platforms moet kunnen draaien. Die kan je van te voren meten. Dan is portability an sich een non functional die je meeneemt in je product ontwerp

P3: Dus eigenlijk zeg je, hoe cloud independent ben je, terwijl we nog niet over gegaan zijn naar een andere cloud. Maar hoe ga je dat definiëren? Hoe vertaal je dat naar een metric?



P2: Een optelsom maken van het aantal cloud agnostische tools die we gebruiken, zoals Terraform

P3: En wat is dan de baseline? Ten opzichte van wat?

P2: Ten opzichte van zichzelf. Je kan niet zeggen van het doel is tien, en als je er tien hebt dan ben je heel goed. Dat is wel heel interessant

P1: Je hebt een meetwaarde, maar je hebt geen meetlat

P2: Maar het kan wel zijn als je de meetwaarde toepast op duizend producten en ziet, deze heeft een score van twee en deze heeft een score van honderd

P1: Die vind ik interessant. Want stel die ene van twee doet het nog steeds op drie cloud platformen, precies wat die moet doen. Maar in ieder geval niet door gebruik te maken van cloud agnostische. Stel je hebt een heel groot team aan developers, en je maakt bewust de keuze we maken drie implementaties, en ongeacht wat ik ermee doe het draait op een platform. Hij herkent zelf wat het is, en hij kan zelf zeggen: ik doe smaakje A, B, of C. Dan heb je nog steeds je doel bereikt

P1: Of heb je dan drie losse software dingen gebouwd?

P2: Ik zeg niet dat het een goede keuze was. Maar dan is het eigenlijk een meetstaf van je functional suitability onder andere, tussen platformen. Tussen voorgedefinieerde platformen

P1: Als je een game uit brengt en hij is niet compatible met een gameboy dat je zegt lekker boeien, dat maakt niet uit

P2: Dus het is eigenlijk een meting van de verschillen tussen je functional suitability die je gemeten hebt per platform

P1: Dat zou misschien kunnen

P2: En dat is voor functional suitability een, ik denk dat security ook wel interessant is. Want als die niet secure is op de een, \*inaudible\*, denk ik dat dat ook wel relevant is

P3: Dus deze hangt over meerdere andere heen

R: Zou die ook over alle andere heen hangen? En zou je dan een niveau kunnen abstraheren en zeggen: in principe kijk je naar tot welke mate holden alle andere metrics in een nieuwe omgeving. Dat is eigenlijk wat ik jullie hoor zeggen

P3: Het moet reliable zijn, je moet hem nog steeds kunnen onderhouden, je moet er nog steeds goed mee om kunnen gaan, hij moet nog steeds performen, hij moet nog steeds doen wat die al deed, dus ja. Tenzij jullie hem anders zien

P2: Als dat een van de vereisten is van je product wel. Dus inderdaad wat je zegt klopt

P3: Ik schrijf hem op. Niet perse als een metric

P2: Ja, een soort metric over je metrics

R: We kunnen gewoon zo veel mogelijk ideeën opschrijven. En over de verschillende focus groups kijken: hoeveel support is er voor dit idee. Als het dan ook in een andere focus group wordt gepitched heb ik meer relevantie voor dit idee. Dat dit volgens ontwikkelaars een goede manier is om dit te meten. We hoeven dan niet zozeer in te gaan op: hoe gaan we dat daadwerkelijk doen?

P2: Ik denk op het moment dat je die andere meetlaten hebt, dan is het een kwestie van \*inaudible\*. Want als je bijvoorbeeld functional suitability volledig kan kwantificeren, en je kunt dat per platform doen, dan kan je ook het verschil tussen die metingen

R: Goed, zullen we voor time constraints doorgaan naar de volgende? De volgende is reliability en reliability is de "degree to which a product, system or component performs specified functions, under specified conditions, for a specified period of time"

P1: Lekker SMART

R: Heel SMART



P3: Eigenlijk zit het al in de beschrijving. Dus het hangt ook weer samen met wat moet het kunnen, maar je moet weten welke gespecificeerde functionaliteit is het. Onder welke, in welke tijd, en als die dat kan dan is die super reliable

P1: Ik zit te denken aan dingen zoals uptime numbers. Mijn systeem moet ten alle tijden minstens 95% actief zijn

P3: Onder gespecificeerde

R: Zou je stellen dat de baseline hier dan altijd honderd zou zijn? Misschien is honderd niet altijd nodig. Is de baseline dan niet honderd? Hoe bepaal je dan de baseline?

P3: De baseline is de totale tijd die de applicatie aan staat, dus vanaf het moment dat

P1: Ook niet eens, sommige applicaties kunnen prima elke nacht een uurtje off zijn

P2: We hebben nu honderd, maar laten we het niet alleen over percentages hebben. Ik kan mij voorstellen dat het misschien ook mogelijk is om te meten over absolute aantallen. Als ik elke nacht maar honderd gele ballonnen verkoop

P3: Dat zijn die specified conditions toch. Dus als je specificeert dat die elke nacht een uur uit moet gaan, en hij gaat twee uur uit. Dan gaat er een uur af van je condities, dus niet reliable. Als jij zegt, kijk even wat google zegt. 99 komma, 5 negens of 4 negens. Dat is hun reliability USP. En dan kan je achteraf meten. Op 1 januari 2023 hebben we de applicatie gereleased. Aan het einde van het jaar gaan we kijken hoeveel tijd van het hele jaar was die daadwerkelijk running. Hoe reliable was die? Op basis van tijd, maar ook op basis van wat die moet kunnen. Specified functions kan je hem aanmeten, en de tijd

P2: Even los van de condities, wil je dus weten hoe performed die. Je meeteenheid, en hoe performed die over tijd. En gemeten per hoeveel, per tijdspanne. Als je zegt hij moet 99% van de tijd online zijn, over een week. Dat je dan over een week kan reflecteren, 99% van de tijd. Als hij er een uur uit ligt over een uur, dan is die gewoon nul procent online. Maar als je zegt het gaat over een heel jaar

P1: Je moet inderdaad ook een tijdsspanne erbij haken

P2: En je moet misschien ook een penalty geven aan het niet voldoen aan. Een uur offline zijn per jaar klinkt misschien prima, maar voor sommige dingen is dat vreselijk. Misschien een milliseconde per dag, maar als dat nu alleen stuk is is dat niet meer oke. Het is lastig om daar een

R: Zou je er dan meerdere metrics van kunnen maken? Als je deze in conjunction met elkaar meet, dan kan je hier iets over zeggen?

P2: Misschien de verdeeldheid van je downtime, dus hoe gegroepeerd is je downtime? Of meerdere van dezelfde metingen, per dag, per maand, per jaar tijd wil ik zoveel procent hebben. Want ook als je die over je dag meet dan ga je nog steeds zien dat die daarboven moet zitten

P1: Downtime over verschillende tijd ranges

P3: Dus als die offline is, mag die maximaal zolang offline zijn

P2: Dus over zoveel hoeveelheid tijd. We hebben het over percentages. We hebben Het specifiek over downtime, maar het zou ook over absolute aantallen kunnen gaan. Aantal requests wat niet behandeld kan worden, een absoluut aantal

P3: Dus het aantal gemiste?

P2: Je hebt een meeteenheid. Dat is dan een aantal, een percentage, een iets. Je hebt een tijdsspanne waarover je het meet. En je hebt een aantal van je eenheid, wel of niet. Waar je wel of niet aan wilt. Dus in hoeverre je down bent of niet

P3: Aantal wel of niet

P2: Ja, wel of niet behaalde kwantiteit van je eenheid

P1: Er schiet mij nog iets te binnen. We zijn nu puur aan het denken aan systemen die of down zijn of niet, maar misschien is er ook wel iets wat er soort van tussenin kan zitten. Hoe ga je dan weer

P2: Je maakt het moeilijker

P3: We hebben niet veel tijd he

P1: Maar de reliability hoeft niet alleen te zijn, doet die het of niet. Het kan ook zo zijn van: hoe goed doet die het? Zeg maar

P2: Vind je dat reliability of vind je dat performance efficiency? Want wat performance efficiency eigenlijk zegt is in hoeverre maak ik genoeg. Performance relative to the amount of resources used under stated conditions, dus hoe efficient ben ik bezig

P1: Ik zat zelf te denken als iets zoals Discord. Discord kan prima 24 7 honderd procent aan staan, maar als nog steeds iedere dag honderd mensen hun gesprek er uit gaat omdat hun gesprek eruit valt, dan ben je nog steeds niet honderd procent reliable, ondanks dat je uptime dat wel is.

R: Dat gaat dan denk ik over het gedeelte specified functions right? Dan werkt een deel van je functionaliteiten voor een gedeelte van een tijdsperiode niet?

P2: Dat is misschien wel eentje die we moeten opschrijven, je meetwaarden moeten wel aan de juiste. Je moet granulair genoeg meten

P3: Nog een keer

P1: Je meetmomenten en je metingen zelf moeten specifiek genoeg zijn. Daar komen dan honderd duizend verschillende waardes uit, die kun je samen voegen tot een metric

P3: Niet perse een metric, maar wel een opmerking

P2: Was dat reliability of missen we nog iets?

R: Ik denk dat we inderdaad goed hebben gekeken naar de specified period of time. Dus je meet het allemaal binnen een time window. Conditions hebben we bekeken, doet het het wel of doet het het niet, en we hebben kort even gekeken naar specified functions. Dus ik denk dat een interessante waar we nog niet een directe metric bij hebben, een deel kan het wel doen en een deel niet. Die noemden we net al. Zou daar ook nog een metric bij horen, of denk je dat dat ook gecovered is in dit?

P2: Je kan meten hoeveel van je platform up is. Doe je dan een meting over je metingen

P1: Dus als je nu een meting doet, honderd metingen, 99 daarvan zijn up, en een doet het niet. Hoe interpreteer je dat dan. Is dat dan een 1% failure, of niet. Daar kan je ook weer een metric aan vast hangen dan. Over tijd is het natuurlijk belangrijk om in de gaten te houden

P2: Als het belangrijk is voor je platform kan ik dat voorstellen, ik kan mij voorstellen ook dat. Het is wel lastig om daar grip op te houden

P3: Nou als je er meerdere metrics aan hangt natuurlijk niet. Je kan voor die ene function zeggen, deze function moet het zoveel van de keren doen, van de totaal gebruikte applicatietijd, moet die het zoveel tijd doen. Ga je dan kijken naar tijd, of keer

P2: Het doen?

P3: Het doen wat die moet doen, dus functionaliteit vervullen

R: Dus eigenlijk dezelfde metric die je hier stelt voor uptime zou je ook nog granulariteitslevel feature overheen willen meten?

P3: Per specified function, die moet ook iets kunnen vervullen. Ik zet hem er wel bij

R: Zou je die dan in combinatie, of in plaats van deze gebruiken? Want ik kan mij voorstellen dat als er downtime is, dan werkt geen van de features bijvoorbeeld. Zou je ze dan beide gebruiken? Beide is belangrijk. Je kan wel totale uptime hebben van honderd procent, maar als tijdens 100 procent uptime 99 procent van je functions het niet heeft gedaan, dan is het

nog een bouten applicatie, om het even plat te slaan, sorry. Dus een granulaire. Ik kan niet meer denken. Uptime per specified functional, ten opzichte van je gestelde doel

R: Laten we daarna doorgaan met maintainability, anders runnen we wat over de tijd. Oke check. Puntje maintainability. Maintainability is de “degree of effectiveness and efficiency, wederom, to which a product or system can be modified by the intended maintainers”. Wat maakt een product maintainable?

P1: Zo min mogelijk hoeven veranderen

P3: Dit is wel leuk want dit is waar we vanochtend ook over na hebben gedacht: modulariteit. Als ik iets aanpas, wat is dan het effect

P1: Gaat dit over, stel je hebt beheerders, en die beheerders hoeven niks aan te passen. Is dat dan heel positief, of gaat het over de getallen dat die wel aangepast moet worden

P2: Je kan het beide als een metric opschrijven toch?

P3: Dus hoe maintainable is die? Kunnen we alles aanpassen wat we willen aanpassen? En als we iets aanpassen, hoe makkelijk is het om aan te passen zonder dat de hele boel omvalt

P2: Stel dat de beheerders alles aan kunnen passen wat ze willen aanpassen, dus ze snappen alles heel goed. Maar ze moeten wel elke dag eraan sleutelen, is dat dan een goede meting? Want hij is dan heel maintainable, of zeg je dan: we hebben hem zo gebouwd dat we heel veel moeten aanpassen

R: Wat ik je nu hoor zeggen, zijn er meerdere metrics die elk hun eigen gewicht moeten krijgen?

P2: Dat is de vraag een beetje. In hoeverre moet ik maintainen? Of in hoeverre kan er gemaintained worden op het moment dat het nodig is?

P3: Als ik hem letterlijk vertaal, in hoeverre kan je het systeem modificeren, door de personen die daarvoor bedoeld zijn. Want als ik kijk naar hoe vaak moet ik onderhoud plegen, dan is het meer reliable. Dus hoe betrouwbaar is je applicatie

P2: Doet die wat die moet doen? Functional suitability. \*inaudible\*, noem je dat maintenance of noem je dat doorontwikkeling zeg maar?

P1: Dit is meer, we gaan ervanuit dat iets kapot is, hoe makkelijk is het om het op te lossen

P2: Als het zich voordoet, in hoeverre kan het product, moet ik het goed zeggen. “Can be modified”, door de intended maintainers. Dus je moet eigenlijk al weten: wie gaan het beheren? Welke skillset hebben die mensen? Je kan een product bouwen

P3: Een hele makkelijk is, kan iedereen erbij ja of nee

P2: Wel een goeie, als je erbij kan maakt het niet uit of je allemaal dezelfde programmeertaal kent, of je kan er niet bij. Dus in hoeverre hebben beheerders de toegang en tools om het te kunnen beheren

P3: Heel erg een doktershandschrift, maar goed we snappen het. Toegang, tools en kennis, in hoeverre hebben die beheerders dat

P1: Maar dat is moeilijk te kwantificeren. Wat je wel kan kwantificeren is, hoeveel configuratie parameters heeft je platform. Het is niet dat meer beter is, maar het geeft je wel een getal om eraan te verbinden zeg maar

P2: Ik zit dan ook te kijken, want wat houdt maintenance in. Als het betekent: hij doet niet helemaal goed ik moet wat aanpassen, dus bijvoorbeeld source code. Als ik het doe in een programmeertaal die niemand kent, dan is het heel lastig. Maar wat zijn de scenario's die zich kunnen voordoen, wat moet een beheerder allemaal kunnen doen? Dat is interessant. In hoeverre die daaraan kan voldoen

P1: Dan heb je daar ook weer een metric, hoe lang tot end of life van technieken

P2: Dus eigenlijk is de hoeveelheid tijd waarin die maintainable moet zijn ook relevant

P3: Ten opzichte van?

P2: Nou als je nu iets gaat bouwen in COBOL of FORTRAN, succes ermee. Dan kun je het nog zo mooi maken. Als ik nu iets in Angular ga bouwen, good luck

P3: Dus hoe vertalen we dat dan?

P2: Hoe lang tot end of life. Stel dat ik nu iets ga ontwerpen wat geoptimaliseerd is voor Windows vista, ja leuk

P3: Dan krijg je wel heel veel bevestigingsmailtjes. Hoe noem je dat dan? End of life, van?

P2: De techniek, je platform, je stack

P3: Dus, en dan hangen we daar een score aan

P1: Als je iets bouwt waarvan je weet, de huidige support is nog drie jaar houdbaar, prima. Maar als je nu iets hebt

P2: Je moet iets kunnen hebben ten op zichte van een meetlat, en een meet lat zou iets kunnen zijn als: na oplevering wil ik \*inaudible\*

P1: Bijvoorbeeld, omdat ik de long term support versies van tools heb gebruikt

P2: \*inaudible\*

P1: Dan kom je uit op de vorige vraag die jij net zei: hoe gekwalificeerd zijn de mensen om onderhoud te doen. Kwaliteit en

P2: Ik zit vooral te denken van: wat is het probleem? Is het het kwaliteit van het product, of hebben ze gewoon iemand van straat geplukt die het mag gaan beheren. Dat kan ook, dan is het niet perse dat de product kwaliteit slecht is, maar dan heb je niet de goede mensen

R: Ik denk inderdaad, de standaard gaat over de maintainability van het product. Dus ik denk dat de aanname gemaakt kan worden dat de skillset er wel ligt. Ik denk dat we daar een beetje op af moeten streven, anders kunnen we een hele lange discussie over wat maakt iemand suitable

P2: De disclaimer is dan: we gaan er van uit dat er mensen zitten die de hele techstack kennen, even simpel gezegd

P1: Of dat je dus een soort tool heb gemaakt met een heel configuratie platform er om heen en nog een instellingenpagina \*inaudible\*

P2: Dan moet je dus wel weten, wat moeten beheerders kunnen doen. Wat zijn de KPI's waar je de beheerbaarheid van de applicatie op afrekent

P1: Er zit ook nog een heel groot verschil tussen functional maintenance en technical maintenance. Dat zagen we in Den Haag

P3: In Rotterdam ook

P1: Ja ook

R: Oke, voor tijd constraints: door gaan naar de volgende?

P1, P2, P3: Ja

R: De volgende is security, en dat is de "degree to which the product or system protects information or data, so that persons or other products have the degree of data access appropriate to their level of authorization

P2: Heb je even?

P1: Ik heb een simpele metric: voldoe je aan AVG?

R: Een metric kan inderdaad ook gewoon een boolean waarde zijn

P2: Je hebt dat soort metrics, maar je hebt ook metrics als hebben mensen de juiste toegang tot de juiste resources

P1: Is het admin wachtwoord niet 'welkom'?

P2: Dus zit er een hele ACL achter die ervoor zorgt dat de juiste mensen de juiste rechten op de juiste plek hebben

P1: En komt de ACL overeen met wat je gedefinieerd hebt. Want er zit een ACL achter, en de vraag is of deze voldoet

P2: Hoe ga je dat verwoorden

P1: Die is goed want het gaat niet alleen maar over permissies. Denk maar aan een Rotterdam, of een Den Haag met andere assen waarop je ook wilt beveiligen. Denk aan informatie categorieën waar alleen selecte mensen bij mogen. Dus ongeacht of je permissies hebt op het bestand, want de hele organisatie heeft permissies. Het heeft te maken met gele ballonnen, dus alleen maar gele ballonnen \*inaudible\*

P3: Er is toch ook een hele ISO standaard hiervoor? Voor security. Dus die kunnen we ook meenemen

P2: Ongetwijfeld

P3: Wat is de score voor ISO standaard weet ik veel hoe veel. Ben geen \*naam collega\*

P2: Maar is dat ook een product score ding? Of is dat een organisatie ding?

P3: Ja, volgens mij heb je vast ook een ISO standaard voor product scope

P1: Daar zeg je wel wat, want we zijn nu ook organisatiedingen aan het opnoemen. Maar het is eigenlijk hoe goed kan ik een hele ACL implementeren in mijn product. Dat is al genoeg. Of het fout wordt toegepast, dat staat los hiervan

P2: Eens

P1: Maar dat is niet echt makkelijk te meten

P3: Maar dan moet je wel weten hoe je permissies en welke andere assen je wilt beveiligen of toepassen

P2: Maar dan heb je het specifiek over wie, wat, welke permissies, en niet over andere assen \*inaudible\*. Dus de ACL is een implementatie

P3: Kunnen meerdere dingen zijn? Dit is een metric

P2: Maar het is een overkoepelend ding. Het gaat erom dat je het mogelijk kan maken om de juiste mensen toegang te geven tot de juiste informatie. Op basis van de juiste hoe noem je dat?

P3: Behoefte, least privilege

P2: Nouja, dat verschilt dus per organisatie. Er zijn organisaties waarin men zegt ik wil dat die persoon bij dat ding kan, en dat is goed genoeg. Of een groep ofzo, en die hebben deze rol. Ik wil dat ze kunnen lezen, schrijven en niet meer

P3: Maar dat is logisch want je hebt gewoon security requirements, en voldoen we daar honderd procent aan ja of niet

P2: En er zijn ook organisaties die zeggen dat is niet goed genoeg. Ik wil er nog wat lagen over heen

P3: Dan hebben ze dus meer requirements. Dus pakket aan requirements is je baseline. Dus alles is ten opzichte van de baseline

P1: Degree to which a product or system protects information or data

P3: Wordt het dan ook weer, dan vallen we in herhaling, wordt het dan ook niet weer een checklist met al je security eisen opgesteld door de organisatie. Die een voor een afgaat

P2: Dan wordt het een organisatie specifiek ding, en de vraag is in hoeverre je iets generieks kan maken. Ik kan mij voorstellen dat je wel een algemeen ding wil hebben, en dat er organisatie specifieke dingen zijn logisch. Ik ken de standaarden niet uit mijn kop

P3: Ik zeg organisatie maar kan ook prima voor de software zijn, dus we hebben een lijst aan security eisen waaraan de software moet voldoen. Voldoen we daaraan, ja of nee. En dan zou je weer per onderdeel kunnen kijken of je eraan voldoet. Dat kan je weer optellen van: ik voldoe aan zoveel van mijn eisen. Misschien is er wel een best practice

P2: Stel dat een organisatie zegt: er is een intern product, daar kan je alleen maar in als je op ons netwerk ligt. Dus we vinden beveiliging niet zo interessant. Betekent dat dat jouw product goed is? Omdat je eis is: ik verwacht geen security. En die vind ik wel interessant. Dan laat je het helemaal aan de organisatie over, terwijl het binnen die context zou je kunnen zeggen: het kan geen kwaad

R: Ik denk dat we hier dezelfde kwestie hebben die we met portability hebben gevoerd. De metrics nemen aan dat de kwaliteitskarakteristiek van belang is. Wat natuurlijk kan als je het model omhoog gewichten aan gaat toekennen, dat hier misschien gewicht nul aan toe komt. Dus dat die metrics niet belangrijk zijn. De metrics die we nu zoeken zijn er op gebaseerd: deze is wel belangrijk tot op bepaalde hoogte

P2: Ja

P1: Een concreet simpele: het aantal gemelde incidenten per tijd mag niet groter zijn dan zoveel

P3: Ohja. Security breaches?

P2: Ja, ervanuit gaan dat alles wordt gemeld

P1: Het is zeker niet waterdicht. Maar stel je hebt tien meldingen per maand en normaal heb je er maar vijf, dan is er iets niet oke

P2: Ja precies

P1: Het is op zichzelf geen waterdicht ding. Dat geef ik je

P3: Oke, dan hebben we er weer drie. Volgende?

R: Laten we dat doen, we hebben nog een kwartiertje. Usability. Usability is de "degree to which a product or system can be used by specified users to achieve goals with effectiveness, efficiency, and satisfaction"

P1: Die laatste is dus heel erg subjectief

P3: Exact, want ik wil misschien helemaal geen gele ballonnen, en jij wel

P2: Dan ga je naar gele ballonnen punt nl en dan wil je geen gele ballonnen

R: Waar je naar kan kijken is: hoe uit satisfaction zich? Hoe uit het zich? Je kan natuurlijk aan een gebruiker vragen: hoe satisfied ben je? Je kan natuurlijk de acties die volgen uit satisfaction anders zijn dan wat ze aangeven voor satisfaction. En dat is ook waar we voornamelijk naar gaan kijken

P1: Mijn hoofd gaat naar conversie rate. Bij sommige producten zou de conversie rate zeker een indicatie zijn van satisfaction. Het zal een kant en klare metric zijn

P2: Is dat ook niet een metric voor je efficiency. Stel je moet maar een formulier door in plaats van vier

P3: En wat was die derde nou? Effectiveness, efficiency en satisfaction. Conversie rate dan heb je het specifiek over iets van een webshop ofzo

P2: Bijvoorbeeld, maar je kan het ook hebben op andere plekken. Denk bijvoorbeeld aan het \*naam systeem\* systeem waar wij aan werkten. Hoeveel van de documenten die gezocht werden worden daadwerkelijk gevonden

P3: Dus hoe ga ik dat generiek maken. Conversie rate

P2: Je hoeft het niet specifiek te maken. Afhankelijk van wat voor soort systeem het is komt daar een getal uit rollen

P3: Het is lastig he om deze niet subjectief te maken. Het hangt aan een persoon af, en die vinden heel veel dingen. Maar hoe ga je een bevinding nou objectief maken

R: Hoe uiten ze zich. Met welke acties uiten ze zich hoe ze zich ergens over voelen? Ik denk dat dat voornamelijk de vraag is die we hier beantwoorden

P2: Er staat ook heel specifiek bij specified users. Dus je moet ook heel goed weten wie zijn je specified users. En heb je daar een meetlat voor, of heb je daar andere metingen voor. Ik

kan mij voorstellen als je een product hebt met gebruikers en beheerders, en die beheerders, dat zijn ook gebruikers, maar met een ander doel. Ik kan mij voorstellen dat je daar een tweede meetlat tegenaan houdt.

P3: Kan je de tijd, de totale tijd dat je iets gebruikt kunnen zien als een measure voor usability. Over het algemeen, tenzij iets heel crap is en je juist moet wachten

P2: Of je juist een product hebt waar je zo lang mogelijk bezig mee wilt zijn. Dus afhankelijk wat je product is

R: Er kunnen verschillende metrics zijn. Als je zegt: deze metric is niet belangrijk voor het product dan is het gewicht gewoon nul. Ik zou beide gevallen dan noteren. En dan ga je specifiek per product kijken, hoe belangrijk zijn deze metrics en dan zeg je: in dit geval hebben we een game, je wilt dat die zo lang mogelijk gespeeld wordt. Maar stel dat je op gele ballonnen punt nl zit, dan heb je dat misschien compleet niet. Dus ik zou stellen dat beiden metrics zijn, en bij het toekennen van gewichten je dan kan zeggen: de ene is wel belangrijk en de andere

P2: Aan de andere kant, als ik gele ballonnen punt nl had en iemand de hele dag op gele ballonnen punt nl zat, maar besteld wel elke vijf minuten wat. Dan ben ik daar heel blij mee. Op basis van wat je use case is ben ik daar heel blij mee. Afhankelijk van wat je use case is moet je meetwaarden hebben. En wat die meetwaarden zijn. Ik kan mij voorstellen dat dat heel specifiek is per product

P1: Nog eentje die generiek is: social media tags. Hoe vaak wordt je getagd op social media

P3: Positief of negatief. Positief zou je doen. Sentiment

P2: Voor een consumer achtig product is dat heel relevant. Voor interne producten helemaal niet. En in hoeverre je voldoet aan tools die je helpen om te begrijpen wat er in beeld staat. Dus screen readers, dat soort dingen. Ik kan mij voorstellen dat dat relevant is, in een bredere context

P1: Valt dat onder usability? Zeker. Maar hoe meet je dat. Je compatibility met screen readers, dat soort dingen

P2: Het is eigenlijk een as op je specified users. Dus welke hoe ga ik dat zeggen. Wat kunnen jouw specified users wel of niet. En in hoeverre wil je nog steeds dat ze ongeacht van wat ze wel of niet kunnen en in hoeverre kunnen ze wel of niet gebruik maken van je product

P1: Kleurenblind modus, of dat soort dingen

P3: Zijn daar best practices voor?

P2: Zeker, standaarden die relevant zijn voor je specified users. Dus specified users is meer dan beheerders en eindgebruikers om even zo te zeggen. Maar ook welke categorieën kan ik daarbinnen bedenken. In hoeverre moet ik mijn data pipeline aan een braille pipeline kunnen koppelen

P3: Volgende?

R: We hebben nog zes minuten laten we sneller doorgaan. Compatibility is de "degree to which a product or system can exchange information with other products or systems". Dat is dus de informatiestroom tussen systemen. Wanneer is jouw systeem compatible met andere systemen? Wat kan je doen om jouw systeem meer compatible te maken

P2: Dus in hoeverre in die plekken waar ik informatie uit deel. In hoeverre gebruik ik daar standaarden voor

P1: Of hoe actueel wordt de techniek die daar achter zit geacht. Stel je voldoet aan standaarden voor rooksignalen

P2: Ik weet niet, ik denk niet dat er heel veel connectors voor rooksignalen zijn

P3: Dus maak die eens concreet



P2: Stel je hebt een standaard voor SOAP, je wilt hem algemener. In hoeverre de gebruikte techniek of protocollen nog actueel zijn. En dat zou je ook weer op een of andere manier kunnen meten. Informatie delen gaat het specifiek over he

P1: Zitten we niet te moeilijk te denken, als in hoeveel talen ondersteunt het. Niet zo zeer programmeertalen. Met welke talen kan ik communiceren

P3: Er staat wel other products or systems

P1: Nee ik bedoel ook niet programmeertalen, maar rooksignalen. Taal in de breedste zin van het woord

P3: Je kan hem ook algemener trekken. Aantal mogelijke connecties

P2: Ook hier moet je weten waar je tegen aan meet. Aantal mogelijke connecties. Ook hier moet je weten waar je tegenaan meet. Stel je hebt een product waar een SDK helemaal niet relevant is, dan heb je geen SDK. Maar stel dat je iets hebt dat juist bedoeld is om je andere producten te integreren, dan is het wel relevant

P3: Dus ten opzichte van de wens: hoeveel connecties kunnen we maken

P2: Dat komt weer terug naar wat jij zei R, in dat geval zetten we hem terug naar nul

P3: Soms staat er nul, soms hebben we wel concreet

R: Dat is oke, tijdens het verwerken van de resultaten gaat alles gestandaardiseerd worden naar een concrete vorm. Oke, performance efficiency is de "performance relative to the amount of resources used under stated conditions"

P2: Gaat dit ervan uit dat het product al af is? Of kan een resource ook een developer zijn in die vorm dat je efficiënt om gaat

R: In dit geval bij deze quality characteristics zijn resources geen personen

P3: Dus ik heb een applicatie, die heeft zoveel gig CPU, zoveel RAM. Hoe goed doet die het?

R: Ja, gegeven een set aan resources, dat klopt ja

P3: Die is echt super kwantificeerbaar

P2: Hoeveel stroom gebruikt die

P1: Iets wat mij stiekem van binnen pijn doet. Cyclomatische complexiteit

P3: Dat moet je even in het Nederlands zeggen

P2: Als je zegt cyclomatische complexiteit

P3: Waar haal je dit vandaan

P2: Stel je hebt een. Het gaat over de hoeveelheid iteraties die een product maakt over een stuk code. Stel je hebt zes for loops in elkaar, dan is de kans op een hoge cyclomatische complexiteit heel hoog. Want het hoeveelheid paren

P1: Ja, kansparen dat je hebt

P3: Stroomverbruik, aantal, euro's?

P2: Beide. Maar ook gewoon zoiets als. Je werkt in een GCP zoals wij aan het doen zijn. Hoeveel euro verbruikt mijn platform? Maar ongeacht welke metric je pakt, wat verwacht ik. Wat voor meetlint

P1: Als je zegt, ik gebruik zes terrawatt, maar ik heb wel wereldvrede, weegt dat tegen elkaar op. Wat is efficiënt? Dat is dan de vraag daarna weer

P2: Je moet het überhaupt kunnen meten?

P3: Ik moet gelijk denken aan \*naam collega\* inderdaad. Laatst hadden we het over de kosten van ons platform en toen zat \*naam collega\* mee te kijken en zei hij. Maar wij bij een andere klant verbruiken zoveel per dag aan euro's. Dan is dat van jullie peanuts. Terwijl bij ons de performance efficiency misschien niet zo goed zou zijn, voor een lagere verwachting

R: Ik denk dat dat exact de conditie is bij performance efficiency

P3: Ja, wat verwacht je en wat is het. Dus hoe efficiënt is het

P2: Je hebt hier wel standaarden voor. Cyclomatische complexiteit is er eentje die eigenlijk heel slecht is

P1: Er is iets als database interacties. Database leesschijf

P2: Weer hetzelfde toch. Wat verwacht je, wat is normaal. Voldoe je aan normaal? Wat wij nu bouwen, dat zijn vrij veel database connecties, daar wil ik wel wat aan doen. Omdat we niet extreem veel database connecties zullen verwachten. Kunnen we dat kwantificeren? Ja, hoeveel we er wel verwachten en wat er tegen kunnen doen

P1: Percentage CPU, RAM, allocated resources

P2: Dat is ook een interessante

P3: En dan tegelijkertijd ook, hoe vaak lopen we tegen de top aan. Percent of time

P2: In hoeverre gebruik je überhaupt de resources die je hebt. En indien je aan de boundaries zit van je gebruik, wat zegt dat. Nul procent is eigenlijk het eerste scenario.

Honderd procent betekent je hebt niet genoeg resources of je bent super inefficiënt

P1: Klopt, en hoe goed kun je het up en downscalen

P2: Ja, indien dat kan ja klopt. Maar dat is eigenlijk ook een metriek daarvan. Want als je \*inaudible\* betekent dat misschien dat je usability niet goed is.

P1: Misschien als dat een functionaliteit is van je product

P2: Ja, maar dan komt er zo'n ouwe vent met een snor en 128gb ram wel even bij prikken. Verticaal schalen, check

R: Oke, ik denk dat we nu wel op de tijd zitten. We zitten er zelfs drie minuten over heen. Ik wil jullie in ieder geval super bedanken voor al jullie input. Ik ga de resultaten verwerken en wanneer de thesis resultaten heeft zou ik die ook even met jullie delen. Super bedankt jongens. Dan ga ik de recording uitzetten

P3: Mocht je bij verwerking nog vragen hebben dan weet je ons te vinden