

R: Dank jullie wel, dat jullie deze sessie met mij willen joinen. Deze sessie gaat er als volgt uit zien. Als eerst geef ik jullie een introductie van mijn onderzoek, wat we precies gaan doen en dan zal ik de taken uitleggen. Dus het onderzoek wat ik ga doen richt zich erop om kwantitatieve waardes te vinden voor abstracte concepten die software product kwaliteit beschrijven. Dus in mijn onderzoek heb ik een aantal modellen met elkaar vergelijkt en uiteindelijk tot een meest geaccepteerd model gekomen voor het bepalen van software product kwaliteit. Het is belangrijk dat het hier om de kwaliteit van het product gaat en niet om de code. Oke? Daar is uitgekomen de internationale standaard ISO 25010, ISO 25010 bestaat uit drie niveaus. Op level een hebben we software product kwaliteit. Op level twee hebben we acht kwaliteitskarakteristieken, dus acht karakteristieken waar goede software producten aan moeten voldoen. De techniek laat even aan de wensen over. En op niveau drie heb je nog een aantal sub-karakteristieken. We hebben hier bijvoorbeeld functional suitability, die kan weer aan de hand van bijvoorbeeld. Functional suitability bepaalt in hoeverre voldoet je product aan requirements, dus in hoeverre zijn de requirements appropriate is bijvoorbeeld een sub-karakteristiek daarvan. Waar mijn onderzoek zich op gaat richten is om concrete objectieve measures te vinden voor deze acht karakteristieken. Belangrijk om te weten is dat ik specifiek op zoek ben naar objectieve measures, want andere onderzoeken hebben zich gericht op subjectieve measures. Om daar een beter voorbeeld van te geven: stel voor dat je gaat kijken naar de kwaliteit van een auto. Wat bijvoorbeeld een subjectieve measure zou zijn, is bijvoorbeeld in hoeverre ben je tevreden met de auto. Dat is iets wat self-reported is, dat is iets wat iemand zelf zegt. En een objectieve measure zou bijvoorbeeld zijn: hoe snel gaat de auto van nul naar honderd. Alright, is voor jullie het verschil tussen objectieve en subjectieve measures duidelijk?

P1: Ja

R: Oke, de taak die we vandaag gaan doen. Jullie zijn de acht kwaliteitskarakteristieken. Ik ga een definitie voorlezen en dan is het aan jullie om een discussie te voeren hoe we dit karakteristiek objectief kunnen meten

P2: Is het goed als ik wat stoelen uit mijn kamer haal?

R: Ja tuurlijk

P2: Top, pak ik die even snel

R: Is voor nu de ondersoeksofzet duidelijk

P3: Ja zeker. Een vraagje. Is dan die sub-karakteristieken hiervan: dat zijn eigenlijk niet, dat zijn niet indicatoren al

R: Nee, de sub-karakteristieken, wat een deel van het onderzoek ook doet is dat een deel van de developer perceptie overeen komt met de voorgeschreven sub-karakteristieken. Het kan ook zo zijn dat we uit komen op measures die slaan op geen van de sub-karakteristieken, wat zou betekenen dat er een extra tussen niveau moet komen. Dus dat klopt inderdaad

P3: Helemaal goed, ik kan hier zitten

R: We gaan ook notes opschrijven, dat is in ieder geval de taak. Jullie gaan de discussie met elkaar voeren. Voor mij als onderzoeker is het belangrijk dat ik alleen deze discussie modereer. Ik kan zelf niks toevoegen want dat zou de onderzoeksresultaten beïnvloeden. Ik zou dan jullie gedachten beïnvloeden en dat mag natuurlijk niet voor een onderzoek. Dus het zou voornamelijk een discussie tussen jullie worden. Ik ga in ieder geval voor elk van de acht een definitie voorlezen. De definitie is in het engels. De measures mogen zowel in het nederlands als engels gemaakt worden want ik zal hierna nog een vertaalslag doen. Oke? Ik lees hem voor. Ik heb hier een aantal post-its en een pen. En jullie kunnen ze dan bij de karakteristieken plakken. Oke? Dus ik denk dat het na de eerste wel voor zichzelf spreken

hoe het gaat. We hebben dus functional suitability, en functional suitability is de degree to which a product or system provides functions that meet stated and implied needs. Hoe zou je dit mogelijk kunnen meten? Als er nog verdere uitleg over de definitie nodig is laat het ook weten

P4: Heb je een voorbeeld hoe je dat?

P2: Dat mag dus niet, hij mag geen voorbeelden geven

P4: Ohja

P2: Als je een auto designed en hij moet rijden, dat heeft hij zijn function doel behaald

P3: Ik zou dan denken dat het heel belangrijk is hoe de requirements geformuleerd zijn. En dan kan je per requirement erbij langs gaan of het behaald is. Maar dan ligt het eraan wat voor requirements het zijn. Als het harde requirements zijn die meetbaar zijn. Van het programma moet zo snel zijn en in vijf seconden moet het opgestart zijn bijvoorbeeld, dan kan het een ja of nee zijn. Maar als het moet zijn, een bepaalde feature moet een klant tevreden mee zijn, dan is het lastiger meetbaar

P2: je hebt zo die SMART doelen die je kan opstellen voor een product. Of iets measurable is, of dat soort dingen

P3: Ja

P2: Dus als je zegt of je doelen daar aan voldoen kan je het met een ja / nee vraag beantwoorden. Maar als je doelen daar niet aan voldoen moet je een andere meetwijze hebben

P3: Ik denk dat dat ook wel interessant is inderdaad

P1: Ja, en in hoeverre zeg maar als je het echt hebt over, het moet bepaalde klanten bedienen. Als N groot genoeg is, zou een survey toch objectiviteit bewerkstelligen?

P3: Ja, ik denk dat benaderd wel objectief is

P4: Het blijft subjectief toch, als er in een survey iets in zitten blijft er subjectiviteit in zitten. Het blijft de mening van mensen. Al heb je heel veel meningen, het blijven meningen

P1: Het is het benaderen, maar toch met heel veel dingen kan je geen objectieve dingen op stellen

P4: Dus een bepaalde subset van dingen kan je niet vast zetten, is het objectief ja nee. Als het definiëren is

P1: Dat denk ik wel

P2: Maar objectiviteit, is dan moet je niet tevredenheid meten. Maar puur kijken naar eisen en of het daaraan voldoet, en hele klanttevredenheid er buiten laten

P4: Ja

P2: Dus puur voor elke functie die je van te voren hebt beschreven of in ja nee vragen of het te meten is, of in een soort schaal met andere eisen

P1: Maar mag je er dan ook vanuit gaan. Krijg je het eisenpakket gegeven, of krijg je de mogelijkheid om daar je zegje over te doen, want dan kan je herschrijven volgens die smartformulering die je net noemde, maar stel je voor je krijgt een aantal eisen, dan is dat outside je eigen verantwoordelijkheid, of je eigen mogelijkheid

P2: Ja, dan zou je eigenlijk al voor je het product maakt moeten hervormen naar doelen die wel meetbaar zijn, en dan kun je checken. Dat is wel de manier om het objectiever te maken maar dat is wel lastig

P3: Maar de definities van wat je requirements zijn misschien

R: Dit kan inderdaad een objectieve measure zijn, want wat de volgende stap in het onderzoek is, is dat ik aan de hand van alle opgeschreven measures wat prerequisites ga opstellen, dus wat voor documenten, wat voor informatie, wat voor systemen zijn er nodig om deze metrics te kunnen verzamelen, dus dit zou inderdaad een metric kunnen zijn, en

het informatiedoel wat er aan hangt is een lijst van SMART geformuleerde requirements, dus dat mag je aannemen

P3: Oh ja

P1: Oke

R: Ik hoop dat dat helpt

P1: In dat geval, ik weet niet wat dat SMART is, maar klinkt als iets wat objectief

R: Specific, measurable, attainable

P2: Reachable? Realizable?

P4: Testable, mogelijk

P2: Iets met concreet

P3: Ik denk dat een deel van functionele, het gaat om functionele eisen. Een deel is hard meetbaar, of een bepaalde functie er is of niet. Maar een deel blijft denk ik subjectief, of denk je dat je alles wel kan ondervangen in iets hards

P1: Ik zelf dus niet, wat ik net al noemde. Dan kun je het benaderen denk ik, door N zo groot mogelijk te maken

P2: Maar dat is lastig als je 1 product maakt voor 1 klant, is die klant tevreden of niet. Dan kun je die N niet groter maken, dan is dat lastig

P3: Je kan ook kijken naar, niet de N, dat je zoveel mogelijk respondenten hebt, maar of je zoveel mogelijk meetpunten hebt, wel harde dingen. En die drukken dan die zachte dingen uit. Een benadering van tevredenheid op functioneel gebied. Dus wat zijn voorbeelden van functionele eisen, kan je dat geven?

R: Een voorbeeld van een functionele eis kan bijvoorbeeld zijn, de gebruiker moet de mogelijkheid hebben om zijn wachtwoord te veranderen, dat is een functionele eis

P1: Misschien is het wel allemaal objectief te definiëren, misschien kun je alles subjectief ook weer. Je kan objectief meten als hij een voldoende geven. De klant moet het een zes geven, de klant geeft het een zes, dan kan je het objectief zeggen, hij kan een zes geven dus dan hebben we het gehaald.

R: Ik denk wel dat een goede is om hiervoor te bedenken. Geef bijvoorbeeld een voorbeeld van een auto wederom. Een subjectieve eis kan bijvoorbeeld zijn. Een klant geeft aan een tevredenheid te hebben van een acht met de auto. Duidelijk subjectieve eis. Maar goed het kan natuurlijk zijn dat die de auto een acht rate, hij heeft vervolgens de keuze tussen die auto en een auto van zijn vrouw, en in 9 van de 10 gevallen toch de auto van zijn vrouw kiest, ookal zou die die minder satisfaction geven. Dan is er dus duidelijk een verschil tussen de self-reported metric, en de acties. Dus ik denk dat objectieve metrics meer kijken naar acties.

P3: Ja

P4: Maar hoe werkt dat in de context van software dan?

P1: Ja ik snap even niet je voorbeeld

P3: Nou in de context van software kan je bedenken van als ik m zo vertaal, als je je wachtwoord kunt veranderen. En een klant denkt ik wil mijn wachtwoord veranderen, en het punt dat die het ook echt doet. Dan is het soort van geslaagd. Daar tussen zit de hele zoektocht naar die functie, of het lukt of hoe makkelijk het is, of die drie clicks heeft of 10 clicks

R: Een voorbeeld in de context van software kan ook zijn dat de klant zegt de applicatie gebruikersvriendelijk te vinden, maar er vervolgens wel vijf minuten over doet om een makkelijke functionaliteit die in tien seconden voltooid moet zijn te voltooien

P1: Maar dat ligt dan toch aan de functionele eis, als jouw ding was het moet te doen zijn. Of binnen 10 seconden of 5 minuten, dan maakt het voor het behalen van de functionele eis toch niet uit

P2: Als we dat dan weer definiëren als een SMART doel en zeggen, je moet deze functionele eis binnen tien seconden vinden

P1: Dat zou natuurlijk kunnen, maar dat is toch een andere eis

P3: Ja precies, dan maak je het in een ja of nee. Maar je kan die net zo goed in een schaal zetten, of in seconden uit drukken. In hoeveel seconden is degene klaar met wachtwoord veranderen. Je kan bijvoorbeeld het aantal clicks tot een bepaalde functie voltooid is. Of het aantal clicks, als objectief

R: Oke, schrijf hem maar op

P3: We mogen er meerdere bedenken

P2: Ik denk aantal clicks toch, tot een actie voltooid is

P1: Maar is dit niet een hele specifieke? Ik kan er wel 1000 bedenken, is dit niet een hele specifieke voor wat we nu in mind hebben? Omdat we nu toevallig aan die wachtwoorden denken. Op dit meta level kan ik er wel 1000 bedenken. Is dit wat we willen? Ik dacht dat we een meta niveau hoger gingen toch. Het definiëren in een ja nee vraag

P3: Dat is de vraag, je kan ook zeggen dat als je er duizenden hebt, dat je dan, dat is het meta ding dat we dan hebben

P1: We gaan nu dingen zoals clicks, maar als je een webapp hebt, hij moet binnen zoveel seconden zijn opgestart, het moet in zoveel landen beschikbaar zijn. Maar ik denk niet dat dat relevant is voor de scriptie van <naam researcher>

P2: Ik denk dat dit er wel 1 is die het subjectief benadert. Opstarttijd is altijd wel objectief, het is wel iets om om dat subjectieve heen te gaan

R: Ik denk dat in dit geval, er is geen goed fout. Er is ook niet een bepaald abstractieniveau. Omdat we juist geïnteresseerd zijn in de perceptie van ontwikkelaars, maakt het eigenlijk niet veel uit. Als we functional suitability zouden moeten meten, wat voor concrete datapunten zouden we daarvoor moeten gebruiken. Dat is de vraag

P1: Dat zou er een kunnen zijn. Maar ik zat zelf even, we kunnen er zo mee door gaan, even te denken stel we hebben dus gered op die objectief ja nee vragen. Heb je ook een stap verder, het moet countable zijn. Want als je een hele lijst met die measures hebt kun je basisstatistiek berekenen, de average, hoeveel je er hebt gehaald. En dan kun je een project ook vergelijken met een ander project

P3: Omdat je dan een lijst hebt aan SMART doelen

R: Wat je dan hebt is je hebt een lijst aan waarnemingen en vergelijkt met een baseline waardoor je tussen producten

P1: Ja, of tussen teams. Het ene team zit altijd tachtig procent, en het andere team doet er een jaar over. Misschien werken zij op een verkeerde manier samen

P3: Ja, dan heb je hem wat algemener zeg je dan. Niet specifieke doelen opschrijven

P1: Ja, en meer algemeen, en je kan er dan meer basisstatistiek mee doen

P3: SMART doelen, en bedenk er dan zo veel mogelijk. Bedenk er zo veel mogelijk bij jouw product

P1: En dan moet elk doel ook een weight hebben. En dan kun je meten hoe succesvol is het project ten opzichte van een ander project

P3: Maar dan moeten die doelen die je zetten weer toepasselijk zijn op een ander project

P1: Ja stel je voor dat je een weight hebt, en dat is in bepaalde zin wel subjectief. Maar bijvoorbeeld het bouwen van logging, dat zou makkelijk zijn dus dan wil je 1. Het bouwen van, weet ik veel wat, een automatische vertaling is weight 100

P2: Of die functionaliteit werkt toch? Daar hangt de weight aan

P1: Exact ja, en dan blijft het nog steeds SMART

P3: Die weight, zoals je hem nu beschrijft klinkt alsof hoe er gepokered wordt in SCRUM. Er wordt altijd gepokered voor een bepaalde functionaliteit, of een bepaalde feature. Dat is eigenlijk door het geboden van. Zoveel punten voordat het gaat kosten. Vaak in uren, officieel mogen het geen uren zijn, maar mensen denken ik ben hier wacht uur mee bezig. Dat is dan de weight van een bepaalde feature.

P2: Het kan lang duren om te bouwen, maar niet perse belangrijk zijn

R: Dit gaat inderdaad over het product, en niet tot het proces

P3: Maar dan zou je, ik weet niet of het dan nuttig is, maar je zou mensen, een PO ofzo, kunnen laten inschatten die weet

P1: Wat is een BO?

P3: PO, product owner

P1: Ah oke. PO

P2: Je kan dat poken wel gebruiken om wat belangrijk is voor het product, en dan een losse pokersessie hoeveel het gaat kosten. En dan wat jij zegt heel concreet maken en dan met weights eraan hangen. Dan heb je al een concrete measure

R: Dus als we die zouden opschrijven hoe zouden we die dan kunnen formuleren? Dus eigenlijk wat we zouden doen. Voor elke feature zou je een formule applyen?

P1: Ja, stap 1 is het schrijven van die functie eisen in een objectieve ja nee vraag. Daaraan verbind je een weight

P2: Je zei functie eisen beschrijven

P1: Als SMART, ja / nee

R: En vervolgens heb je nog de vraag. Is de functie eis geïmplementeerd, is de functie eis niet geïmplementeerd. Om er een score te krijgen

P2: Je zou dan als SMART doel beschrijven, er een weight aan hangen. En als die behaald is dan is die weight die mee telt aan het eind

P3: Ja, maar ik denk dat we bedoelen is het specifiek of niet. Of is het behaald in dat het in zoveel seconden opstart. Dat is wat je bedoeld met

R: Maar is dat functional suitability of is dat usability?

P3: Nee dat gaat eigenlijk over iets anders. Functional gaat meer over is het er

R: We kunnen uiteindelijk ook post its wijzigen tussen de verschillende attributen. En dat we uiteindelijk bedenken, hij past toch beter bij een andere, want deze dingen zijn dependent op elkaar

P1: Ja, ik zie mij nu al voor mij dat er veel overlap kan zijn

R: Het kan ook dat er 1 metric kan zijn die gebruikt kan worden voor meerdere kwaliteitskarakteristieken

P1: Maar ja, over dat usability. Als iets niet werkt, dus je hebt een login dat niet inlogt, is het er dan?

P3: Is het dan functioneel gehaald, en niet usable, of is het functioneel niet gehaald, wel geen usable?

P1: Exact, in mijn ogen kan je het niet als een gehaalde functionele eis beschouwen

P3: Als je niet kan inloggen. Dus eigenlijk een score 0 bij usability betekent ook een score 0 bij functional

P1: Zou ik zeggen van wel ja

P3: Maar die dingen kan je in een modelletje aan elkaar koppelen

P1: Dat mag <naam researcher> doen

P3: Dat mag <naam researcher> doen

P1: Een PDD. Dat verwachten we wel <naam researcher> Een PDD met 5 pagina's

P3: Meer een statistisch modelletje denk ik dan. Een AND, of zoiets

R: Oke, denken jullie dat aan de hand van deze dingen, dat je functional suitability volledig kunt benaderen. Als het antwoord ja is gaan we telkens door naar de volgende. Ik denk dat dit een beetje de procedure is

P3: Ja volledig is, zeggen dat je de hele ware soort god hebt

P1: Je hebt hier vier beta's <naam researcher>

P3: Nooit echt volledig denk ik dat mogelijk is

R: De best mogelijke benadering die we voor nu kunnen bedenken

P3: Ja, ik vind het niet gek

R: Laten we dan doorgaan naar performance efficiency. Performance efficiency is de performance relative to the amount of resources used under stated conditions. Dus de performance van het product, relative aan de gebruikte resources. En resources kan bijvoorbeeld CPU utilisation zijn. Dat soort resources. Belangrijk om te weten is dat mensen zijn hier geen menselijke resources

P1: Oke, dus niet resources die zijn gebruikt om het te bouwen. Zeg maar developer uren

R: Developer uren, nee

P3: Kan je een voorbeeld noemen, wellicht met de auto. Zodat we niet teveel gestuurd worden

R: Ja, bijvoorbeeld, hoelang rijdt je auto op bepaalde benzine. Als je lang zou rijden met weinig benzine zou het goed raten op performance efficiency. Als je heel vaak zou moeten tanken zou het slechter raten. Want de performance zou dan slechter zijn onder een gegeven resource, in dit geval benzine

P3: Oke, dus dat is die

R: Relative to the amount of stated conditions

P3: En die stated conditions is dan twee liter benzine die je net noemt

P1: Wederom die meta levels. Je kan makkelijk op een laag meta niveau die CPU, computations. Maar dan heb je een hele lange lijst denk ik weer. Maar hoe ga je dan een level hoger

R: Kan nog steeds dat je een lijst hebt met hele dingen. En als je deze set met elkaar meet dan heb je een vrij exacte benadering. Dan kom je in de buurt

P4: Dus geheugen gebruik, CPU utilisation, alle andere dingen

P1: Misschien in dit geval kan het ja

P4: Ja inderdaad. Dan is het *inaudible*

R: Het kan inderdaad dat het bij sommige wel goed kan en bij andere niet goed kan. Dat is ook een onderzoeksuitkomst dat we er bijvoorbeeld achter komen dat het helemaal niet wensable is om het te meten, en dat bij sommige de benadering iets exacter is. Dat is ook nog steeds een uitkomst.

P3: Bij performance kan je wel, een afdoende lijst. Een volledige lijst komen. Network, memory, die Van Neumann computermodelletje, kan je die vijf onderdelen. Je kan netwerk, processor, geheugen, opslag. Dan heb je. Die kan je allemaal langs

P4: Ik denk dat het relatief straightforward is met wat je kan meten. Het is gewoon heel meetbaar

R: Laten we ze dan opschrijven, en als we het niet eens zijn met een

P1: Oke dus werkgeheugen

P3: Hoeveel werkgeheugen kost het

P2: Maar dit zijn allemaal resources, we moeten even kijken is die performance dan wat we in die vorige hebben bepaald. Hoe die functioneel performed. Of hoe meten we dan performance

R: Wat we nu voor de metric nog missen is een baseline. Want je kan wel zeggen een bepaald product gebruikt zoveel gig aan RAM. Maar wat zou de baseline dan zijn. Ga je kijken naar industriestandaarden, ga je kijken naar. Waar zou je naar kijken als baseline. Ik denk dat daar voornamelijk de complexiteit bij performance efficiency bij in zit. Het kan zijn dat je daar een prerequisite voor nodig hebt. De makkelijkste die over het algemeen, is het vergelijken met het gewenste resultaat. Dus het kan zijn dat het dat is. Dus om dit model toe te passen dat je eerst iemand moet gaan vragen: wat is het gewenste resultaat en vervolgens perform je die meting. Misschien ook niet, en ga je naar industriestandaarden kijken

P3: Gewenst is wel nice, maar het is heel erg subjectief omdat jouw objectieve measure af hangt wat iemand zegt als gewenst

P1: Maar hoe ga je gewenst ooit objectief definiëren. Dan moet je twee producten bouwen. Maar als je het gewenste product bouwt dan haal je het al niet zeg maar

P4: Je moet een ijkpunt hebben. En als het dan in de industrie, iets wat vaak gebeurt, dan zit je al bij of bij iemand die zegt, het moet er aan voldoen. Dus ergens een gemiddelde bij halen

R: Een stated condition kan ook bijvoorbeeld budget zijn

P3: Ja natuurlijk, want je kan er heel veel clusters tegenaan gooien. Heel veel grote clusterse

P1: Ja, maar hoe meet je het dan? Stel je hebt een budget van tien euro

P3: Ja, dat weet ik ook niet echt

P1: Maar dat is denk ik wel een fundamenteel probleem. We kunnen het makkelijk opschrijven, we gaan het doen met benchmarks. Maar in een heel custom made product,

stel je wilt vijftien jaar geleden Adyen oprichten, er bestond geen Adyen, wat wordt dan je

P3: Ik zat even te denken, denk met mij mee. Als je bepaalde levels van meetpunten. Dus je meet bijvoorbeeld: hoeveel geheugen neemt het draaien van het hele programma op idle, hoeveel geheugen neemt dat. En dan dat je denkt, bij bepaalde functie, als je dat meet per functie, als je die uitvoert als je dat als meetpunten neemt. En dat in algemene zin noemt. Het hele programma is een meetpunt, dat je bepaalde functionaliteiten die bij elk programma gelden gaat meten. Daar zat ik aan te denken

P2: Ja dan zou je het een beetje kunnen vergelijken met die login, dat zou zolang moeten duren

P3: Ja, dan zou je elk programma naast elkaar kunnen leggen, en kunnen meten

P1: Ja ik denk dat bijvoorbeeld voor een login dat dat zou kunnen. Maar tegenwoordig, een login ga je niet meer bouwen. Alles waarvoor je echt z'n heel uitgebreide analyse gaat doen denk ik niet dat het bestaat

P2: Dan maak je iets nieuws wil je zeggen

P1: Ja, persoonlijk denk ik dus dat voor alle dingen waarvoor iemand moeite gaat nemen om <naam researcher> zijn hele paper te lezen, denk ik niet dat er benchmarks zijn

P3: Maar het is wetenschap, je hoeft niet over de toepassing na te denken

P1: Nee, maar dat is dan toch wel. Dat maakt het wel. Dat is toch een naar idee. Het hele idee is dat je het wel echt. Als je het wilt doen, moet je het ook daadwerkelijk kunnen uitvoeren

P2: Misschien is dan de oplossing om vanuit het budget een aantal eisen van de klant een inschatting te maken wat die resources gaan kosten. Of wat het maximale gebruik gaat zijn. En dan evalueert of dat uit gaat komen en of het je performance gaat verminderen om dat

nog te halen. Dat je dat van te voren vaststelt. En het kan dan zijn met eisen van de klant, dat je het daarin meet. Niet optimaal, maar misschien

P4: Als die er is, en anders een klantvoorstel. Als het niet anders kan. Ik denk dat je haast wel moet

P3: Ja, dan kom je toch weer uit op iets subjectiefs, maar het is lastig om

R: Het is lastig om te zeggen, wanneer is een CPU utilisation van 30 procent goed, en wanneer fout. Ik denk dat dat het lastige is waar we naar op zoek zijn

P1: Ik denk dat je ook kunt zeggen. Heel vaak boeit het niet, want ik denk dat heel veel bedrijven deze eigenlijk niet heel veel. Ik denk dat als je het hebt over green computing, als het belangrijk is en als het al staat. Ik denk heel vaak als dit er toe doet bestaat er al iets en wil je dat het minder resources gaat kosten

P3: Want dan ga je een beetje vergelijken

R: Wat ik jou een beetje hoor zeggen kan natuurlijk ook. Als je een product iteratief wilt verbeteren dan eigenlijk dat je baseline je vorige meting is

P1: Precies, dan gaat het sowieso

P3: Als je het vergelijkt met de vorige versie van je systeem

R: Als jullie erop uitkomen dat dat de manier is, dan is dat ook een uitkomst

P4: Als je het de eerste keer doet, tast je in het duister. Kan het wel goed. Doe dan je best effort, en daarna zien we wel

P1: Ik zou eerst die vijf dingen nog eens opschrijven. Even kijken welke mis ik nu. Oh ja gewoon, harde schijf, CPU. Ik heb nu werkgeheugen, CPU, netwerk, harde schijf. Dat is het wel toch?

P3: Dat is het wel, eigenlijk. Ja, dit is het eigenlijk wel. Je kan zeggen, CPU, hoeveel procent, en hoeveel CPU's in totaal. Dan heb je een tweede erbij. Maar origineel heb je er nog eentje van communicatie in een computer wel. Hoeveel communicatie

P1: Maar dat is netwerk toch

P3: Ja, in een groter netwerk gaat het om. Dus dat is het wel

P1: En dan voor de rest, eerste keer kan niet. Of eerste keer

P2: Best effort

P3: Maar wat voor aan punten geef je dat dan. Wordt het dan waarde 1. Het moet een waarde krijgen zodat de volgende op basis van die

P1: Ja best effort is op basis van die in feite toch

R: Ik denk ook dat een eerste meting. Een eerste meting is niet het meest useful. Een tweede meting, wanneer je een maand later kan aantonen dat je product is verbeterd ten opzichte van vorige keer. Ik denk dat dat is wat mensen interessant vinden

P3: Ja

P1: Ja

P4: Dus dan is het waarde 1

R: Je zou natuurlijk kunnen zeggen, je product scored een 0.7 uit of 1 maar dat zegt in feite nog niks

P1: Misschien is dat inderdaad het hele ding, ten opzichte van een gelijknamig product. En dan is dit perfect

P3: Maar hoe formuleren we het dan?

P1: Maar dan hoeven we het niet op te schrijven en is het meer een general note in totaal. Als hetzelfde met die eerste is ook. Je hebt 80 procent van je doelen behaald. So what? Als je het niet kunt vergelijken met een ander iets, dan schrijf ik dat nu niet op

P3: Dan is het inderdaad dit. En dan moet het wel onder dezelfde CPU, of dezelfde configuratie

P1: Dat heb je wel, CPU uren, dat is wel echt een ding

R: Is onze benadering zo compleet mogelijk voor performance efficiency? Oke, dan gaan we door naar reliability. Reliability is de degree to which a system product or component, performs specified functions, under specified conditions, for a specified period of time

P3: Ja, dus eigenlijk of het niet uit valt

R: Dus in hoeverre performed die nog steeds bepaalde functionaliteiten als je de operational environment op een bepaalde condition stopt. Dus bijvoorbeeld een bepaalde condition zou een hoeveelheid requests kunnen zijn. En een specified period of time kan dan zijn de degree to which de logging functionaliteit nog steeds werkt als je gedurende vijf uur elke seconde 200 requests stuurt, bijvoorbeeld

P2: Dat zou je net zo kunnen benaderen als die eerste. Dat je voor alle functionele eisen SMART doelen opstelt en dan specificeren wat de eisen zijn waaraan die moeten voldoen, aan welk tijdschema, en dan testen, en een score meegeven

P1: Ja, zou ik ook zo zeggen ja. Hier zitten, je vraagt aan de klant of aan een business team. Waar moet het aan voldoen, waar moet deze load aan

P2: Nou dat ging snel

R: Oke. Door naar de volgende? Meer remarks hier over?

P4: Nee, hier sluit ik mij wel bij aan, dit framework

R: Oke, maintainability. Dat is de degree of effectiveness and efficiency to which a product or system can be modified by the intended maintainers. Effectiveness and efficiency to which a product or system can be maintained by the intended maintainers. Hoe zeg je nu eigenlijk of een product maintainable is?

P1: Volgensmij heb je van die libraries die complexity van code meten. Functies, for loops. In ieder geval voor Python ken ik hem

P4: Hoe heet die?

P1: Ja, als. Ik weet de naam niet, maar ik weet dat bij <naam organisatie>, het bestaat wel. Dan kan je op een hele basale manier de complexiteit benaderen

P3: Dat is niet slecht. Ik denk niet dat er 1 kan zijn. Je kan het ook, maar dat is minder hard. Hoe lang het duurt om een feature opnieuw te bouwen. Maar dat is niet heel erg

P1: De grootte van een community van een taal. Als in als je in een soort dode taal zit, bij <naam organisatie> heb je zoiets uit 1980, en dan hebben ze developers van echt 1000 euro per uur nodig omdat niemand die taal meer kan. Bij <naam organisatie> ook trouwens, daar heb je ook die legacy

P4: Dat is nog steeds subjectief

P3: Ja als we het objectief verwoorden

P2: Opzich is het objectief toch, in hoeverre zijn de middelen die je gebruikt nog

P4: Relevant

P3: Hoeveel gebruikers er nog van die taal zijn

P1: De groei van de taal. Dat is niet gek eigenlijk. Of de Reddit

R: Het kan inderdaad een indicator zijn, dus als je een aantal van die indicatoren heb kan je uiteindelijk een exactere benadering hebben dan wanneer je er minder hebt. Dan is denk ik het beste wat je kan doen om een inschatting te maken van deze zogeheten latent variables

P3: Dus wat noemde je als eerste? Dat vind ik opzich wel een goeie

P1: Code complexity

P3: Dat er een bepaalde formule overheen loopt, dat kan sowieso. En dan die tweede hoeveel gebruikers van een taal er nog zijn

P2: Er zijn van die tickets, dat is lastig. Want als er nog geen probleem is geweest met een bepaalde functie kan je de tijd die het kost om het op te lossen niet echt vergelijken. Maar

daar kan je ook niet echt iets over zeggen als je alleen weet hoeveel mensen die taal gebruiken. Dat zegt ook niet heel veel. Als het elke keer twee dagen kost om het te fixen is dat vervelend, maar wil het nog niet zeggen hoe slecht het is

R: Een goede vraag om een metric te bepalen is: wat zou je kunnen doen om de maintainability van een product te verbeteren? Dus op welk attribuut, of welk aspect zou je naar kijken. En ik denk dat je aan de hand daarvan een metric kan opstellen.

P4: Ik zit aan die documentatie te denken, maar dat is zo subjectief natuurlijk. Hoe iets gedocumenteerd is, readme, comments, maar dat is subjectief natuurlijk. Dat is geen metric

P3: Lines of comment. Is niet perse altijd beter

P1: In het algemeen denk ik wel, iets van uitleg bij een functie. Heel vaak heb je geen documentatie, en als er documentatie is. Niemand gaat random lang documentatie. Het is geen uni waar je per pagina betaald krijgt

P3: Dus inderdaad, feit dat het gedocumenteerd is

P2: Dus misschien het aantal functie waar een toelichting op is

P3: Het aantal functies dat wel of niet. Misschien kan je dat tegen elkaar uitdelen

P1: Gewoon docs. Documentated functions

P3: Documented functions delen door niet documented functions. Dan heb je een grotere codebase en heb je gelijk een betere. Terwijl juist de grootte van de codebase vaak wordt dan slechter

R: Ik denk wat jullie heel goed aan het doen zijn is kijken of je bij een metric een baseline kunt gebruiken, of dat de baseline een vorige versie is. Wat we bijvoorbeeld bij performance efficiency heel erg hebben. Dus goed om bij elke metric daar bewust bij na te denken. Want hier zegt het natuurlijk meer om het te vergelijken met het totaal aantal functies. Want als je puur het absolute hoeveelheid met deze, en het absolute aantal. Want er kunnen heel veel functies bijkomen, en ik denk dat dat een hele goede toelichting is

P2: Is dat bij die eerste die we hadden ook. Komt daar ook een score uit?

P1: Nou, wat ik dus weet er was een project gaande. Grote codebase. En ze gingen kijken waar gaan we refactoren, dus ze hebben die library gerund. Daar gaven ze functies uit, en daar is naar gekeken en zijn refactor resources aan gekoppeld. Maar ik was er niet bij betrokken, iemand vertelde er over. En ik denk dat er dus een soort score, ik weet niet hoe hjet precies werkt. Ik denk dat het in depth genoeg is

P4: Ja, ik denk samen zo is het best een goede indicatie van maintainability denk ik

R: Oke, de volgende die we gaan bespreken is compatibility. En compatibility is de degree to which a product or system can exchange information with other products or systems. Dus dit gaat echt over het delen van informatie tussen systemen

P3: Iets van het aantal, wat is het ook. 200 delen door 400 codes. Als je over een netwerk communiceert heb je die HTTP, ik weet het niet meer

P4: Ja, HTTP

P3: Van die bad requests of success. Als je gewoon de geslaagde requests delen door niet geslaagd.

R: Is het sensible om dan te zeggen dat het over informatie exchange gaat?

P3: Het is natuurlijk een protocol, maar niet alle communicatie gaat, gebruikt een netwerk. Wij gebruiken HTTP

P1: Maar misschien ook breder, als jij iets op Azure bouwt, en Azure wordt steeds kleiner en stopt er mee, dan is jou hele ding niet meer compatible. Of games bijvoorbeeld

R: Ik denk dat dat eerder gaat over de volgende, portability. Compatibility gaat echt over informatie exchange

P3: Dus eigenlijk moet je meer denken aan API's

R: Terwijl runt het op verschillende platformen heeft meer met portability te maken, want portability is de degree of effectiveness and efficiency to which a product or system can be transferred from one usage environment to another. Dus draait het op de een, draait het op de ander. Wel een goede. Hier is meer confusion over binnen de standaard

P4: Tussen deze dingen?

R: Ja, tussen deze twee. Het is niet gek dat je

P2: Ja, schrijf je dan die je net noemde op

P3: Maar dat is wel in het geval dat de applicatie met API's wel werkt. Maar dat is uberhaupt compatibility. Stel het is een standalone applicatie

P1: Stel dat iets compatible is via API, via de web. Misschien het aantal vormen van communicatie. Een API is een interface, web is een interface. Kiwi is een interface

P3: Dus wat wil je toevoegen?

P1: Misschien dat dat iets is, het aantal manieren

P3: Het aantal interfaces dat de applicatie is

R: Waarmee zou je dat dan vergelijken? Betekent meer interfaces gelijk dat het beter is?

P3: Ja, misschien moet je aantal interfaces

P1: Ik zou zeggen niet perse, maar dat wordt dan weer gedekt door dat eerste. Door een ander punt. Dan weet ik niet perse welk punt. Het wordt er meer clumsy van. Dan is het niet perse minder usable. Snap je een beetje waar ik heen wil. Dit is misschien er zijn heel veel mogelijkheden. Het is heel compatible, maar dat betekent niet dat het beter is. Maar misschien binnen dit vakje meer compatible

P4: Misschien zo van sommige moeten er zijn, sommige zijn minder belangrijk. Degene die er moeten zijn, moeten er zijn? De functionele eis zeg maar?

P1: Dacht je dat ik dat zei?

P4: Dacht dat je daar op doelde. Niet perse dat het beter is, maar meer dat als je degene hebt die je wil hebben, en die er zijn. Dan is het al gedekt door het eerste punt. Dat het functioneel er is

P1: In dit geval bedoelde ik, in dit punt, hoe meer hoe hoger de score. En als dat uiteindelijk gaat betekenen dat, ik weet niet wat dat gaat betekenen, dat het in een ander punt usability dan

P2: Ja, dat een andere meting lager wordt, en je het toch hebt, dan maakt het niet uit of dit alles omvattend is. Want puur de compatibility geeft dat ook echt aan. Dat vind ik een goed punt eigenlijk, dan maakt het niet veel uit

P4: Sorry

P3: Willen jullie nu iets opschrijven, ik had hem even niet gevolgd

P2: Ik denk dat het wel goed is. We hadden het erover dat het aantal integrations niet een vergelijking ding moet zijn. Want als daardoor de usability wordt beïnvloed zien we dat toch terug in die waarde, dus hoeven we het niet allemaal hier in te vatten

P3: Ja precies, dan is eigenlijk meer altijd beter

R: Is dit wederom net zoals bij performance efficiency kijken naar deze iteratie, en kijken naar de volgende iteratie van het product?

P1: Dat denk ik wel, maar bij elk punt tho. Of denken jullie daar anders over? Zijn er punten waar je het niet met iets anders kan vergelijken?

P3: Geeft het meer inzicht

R: Ik denk dat de grote vraag is, wanneer krijgt een meetwaarde betekenis. Ik denk dat dat de vraag is waar we naar zoeken

P3: Nou, bij sommigen is dat sowieso al. Ja nee ik weet eigenlijk niet

P1: Je hebt altijd toch die baseline is?

P2: Nou je had wel die eerste, met die doelen en als je daar een deel van haalde dan is het op zichzelf al betekenis

P3: Dan heb je al wel iets van een idee

P2: Ja, de rest is allemaal

P3: Maar het krijgt wel nog meer betekenis als je het kan vergelijken met de volgende iteratie, 90%

P1: Maar toch, stel je voor je hebt 80% gehaald. Is dat goed, is dat slecht? Dan heb je ook die baseline nodig toch?

P2: Ja dat ligt dan misschien toch aan de klant, of aan de baseline

P1: Ja, volgens mij zit je altijd met die baseline toch

R: Ja, dus de conclusie die jullie trekken is, deze waardes hebben meer betekenis als je het vergelijkt met een

P3: Met een of meerdere waardes

P1: Ja, met een gelijknamig systeem. Als het heel anders is slaat het nergens op. Stel je voor je wilt een ChatGPT 7 gaan ontwikkelen, dan is dat eerste punt veel moeilijker te behalen dan stel je voor je wilt een website voor een lokale bakker maken. Dan kan je heel makkelijk honderd procent scoren op die eerste, terwijl je hebt altijd een gelijknamige baseline nodig. Of aantal interfaces? Oh die heb je al

R: Oke, is dit een zo exact mogelijke benadering van compatibiliteit. Zullen we door gaan? Laten we doorgaan naar portability. En portability is de degree of effectiveness of efficiency to which a system or product can be transferred from one usage environment to another. Wat maakt een systeem portable?

P3: Iets van aantal uren dat die draait per systeem delen door de errors dat die geeft ofzo

P1: Nou, wat <naam researcher> net zei dat die bij het vorige punt meer hierbij zat, zit ik heel erg te denken aan porten van 1 platform naar een andere

R: Inderdaad, dat is inderdaad het stukje usage environments waar het over gaat. Dus ik denk dat jouw perceptie van de definitie hiervan klopt

P3: Dus ja, in een voorbeeld heb je, van mac op windows

P1: Ja, Linux

P3: Je hebt m gemaakt voor Windows, en nouja je gaat m maken voor een ander systeem. Mac of Windows gaat sowieso niet werken. Je hebt m gemaakt voor een Linux distro en gaat hem op een andere Linux distro draaien. Kijken of die nog problemen geeft

P1: En wat maakt het dan portable?

P4: Of het überhaupt kan, of hoe makkelijk je het kan maken om te porten?

R: Kan beiden

P1: Stel je voor je zit op platform Y en je hebt een miljoen andere computers. Is het dan een definitie van oke je kan naar 80 procent van al die dingen porten, dat is 80 procent portable. Terwijl als je toevallig in een heel oud soort environment zit, een soort Windows XP, dan kan je maar naar drie porten, dan heb je hele lage portability. Zo zie ik het voor mij

P3: Ja

P1: Maar hoe je dat dan preciezer maakt

P3: Maar jij vroeg nu net of je het nog voor je kan zien. Stel je hebt het gemaakt voor XP, hoe lang duurt het voordat je het werkend kan maken voor wat is het, Windows 10

P4: Het is niet of of

P1: Nee fair enough, goed punt. Want als het heel makkelijk is

P3: En daar heb je natuurlijk overlap met maintainability, maar dat is niet ons probleem nu

P4: Ik vind de baseline lastig. Want als je alle lists moet af gaan zijn er een heel aantal

P1: Maar misschien kan je die twee van ons samenvoegen van, je hebt die computer en je hebt een miljoen mogelijkheden. Dus stel je voor je kunt er meteen naartoe, dan is het moeite nul. 80 procent kan je meteen naartoe, 80 keer nul is nog steeds 0. Tien procent kun je met een uur programmeren, dan doe je die. Om het portable te maken voor alles, wat is dan al het productiewerk bij elkaar opgeteld

P3: En dan is minder eigenlijk beter zeg je wel

P1: Ja, absoluut

P3: Ja, hoe ga je dat in een getal zetten

R: Is dit niet dezelfde kwestie die we bij performance efficiency hebben gehad. Dat je het kan meten voor een bepaalde versie van een product, vervolgens gaat een developer werken aan de portability waardoor het makkelijker gemaakt wordt en is die tijd en stuk minder, dus is die tijd. Is dat niet dezelfde kwestie of is het niet wat jullie bedoelen?

P3: Maar wat is daar erg aan? Stel je voor die tabellen

R: Het hele punt is, is de baseline die jullie bedoelen niet een bepaalde versie van het product versus een andere versie van het product. In een eerste versie duurt het tien uur, en in een volgende versie duurt het maar zeven uur. Dus dat je zo die vergelijking trekt

P2: Dat je het iteratief gaat vergelijken

R: Is dat wat jullie bedoelen?

P3: Nou, volgens mij zaten we nu nog niet na te denken over een vergelijking met andere producten toch?

P4: Overhaupt een score toch

P1: Ja, daar zat ik inderdaad over

P2: Het klopt wel

P3: Als je het ene en het andere doet heb je inderdaad de interactie tussen die twee pak je dan

P4: Je kan de score uitrekenen voor een versie van het product, een nieuwe versie maken, nog een keer de berekening maken. Dan weet je ook hoe makkelijk het dan is. Dan heb je een vergelijking tussen versies

P3: Maar hoe komen we op het eerste getal überhaupt

P1: Ik dacht ook dat ik het had maar ik heb het eigenlijk niet. Want daarna een ander ding is, stel je doet het op windows 10. Er zijn nog steeds computers op windows xp. Wordt je portability dan minder omdat het ook weer tijd kost om terug te gaan naar XP, wat je nooit gaat doen

P4: Als je een sample gaat pakken die relevant is, en de rest buiten beschouwing gaat laten

P1: Exact

P3: En die cijfers veranderen elk jaar

P1: Dat is niet perse het probleem. Dat hoeven wij niet te doen, maar stel je gaat ooit die berekening gaat doen moet je dan kijken. Hoeveel XP, hoeveel computers zijn er

P2: Relevante set pakken, daarmee berekenen met hoeveel je compatible bent

P3: In dat geval kan je het definiëren als je berekent elk jaar op het moment dat je het wilt berekenen wat is het aandeel van de gebruikers. Welke systemen zitten ze op en wat zijn de percentages. Stel je hebt 2 procent van de gebruikers gebruikt nog XP, 80 procent heeft windows, en 20 procent mac of een Linux versie. Dat je daar dan mee gaat rekenen. En dan stel het lukt binnen een uur om het naar mac te zetten is dat 20 procent. Weet ik veel 20 uren. Je wilt dan die uren keer

P1: Ik zat daar ook op maar stel je hebt een groot server centrum en je bent zo groot als Google, dan installeer je gewoon Linux op die servers. Dat is opzich geen gegeven dat er een bepaald operating system op staat. Voor ons misschien, omdat we dat niet zo snel

kunnen. Maar als jij echt een bedrijf wilt, bent, en je kunt alles vanuit scratch bouwen. Dan installeer je gewoon zo'n systeem op een computer right

P3: maar dan nog steeds heb je, dan heeft die een lage portability score. Dan heeft dat bedrijf een lage portability score voor die software. Niemand zegt dat dat erg is

P1: Ja, ik vind hem erg lastig. Er komt zoveel bij kijken

P3: Er komt heel veel bij kijken. Je kan sowieso zeggen, hoeveel. Je kan haast niet zeggen op hoeveel systemen die werkt

P1: Nee, want je kunt op schaal van een bedrijf gewoon Linux installeren. Ik heb Windows, ja Mac is iets moeilijker. Hoewel Azure wel. Maar als je AWS hebt kun je met Linux werken en dat is dan gewoon gratis, dan kun je alles installeren

R: Het kan natuurlijk ook dat de conclusie is dat jullie erop uitkomen. Het is het niet waard om portability te meten in dit geval. Dat kan natuurlijk ook

P1: Nou misschien wel, maar is het enige wat je moet meten vendor lock. Dat dan het enige is wat er toe doet

P3: Wat bedoel je dan?

P1: Nou dat je niet vast zit aan Amazon, of aan Azure. Ja dat je daarin helemaal vast zit. Want dat is voor heel veel bedrijven wel het probleem. Dat je binnen een hele dure AWS zit, waar je nooit uit kan want die ene transitie kost gewoon zo veel dat je dat niet bovenop je operatie gaat doen

P2: Is de metric dan, zit je daarin vast of niet

P1: Ja, en hoe duur is het. Misschien hoe duur is het om te transitionen

P4: Dat is een goeie eigenlijk

P1: Hoeveel resources

P3: Dat wordt dan een schatting, dat weet je niet

P1: Maar ik denk met heel veel dingen, dat het een beetje schatten wordt

P3: Ik denk wel dat we die kunnen opschrijven. Bepaalde dingen kun je weten omdat het vast zit in contracten. Misschien kun je beter zeggen. Stel je hebt geen bedrijf, het is software. Dan heb je geen prijskaartje, maar je kan wel manuren zeggen. Kan je beter inschatten dan. Je kan wel zeggen zoveel FTE

P1: Maar dat kost ook weer geld toch

P3: Ja, maar wel uurloon is iets wanneer je zelf een software hebt gebouwd

P1: Ja oke, maar aan de andere kant als je vijf mensen hebt en je hebt in plaats daarvan een senior, wat boeit het bedrijf dan het uurloon. Voor een bedrijf is het toch altijd de kosten.

P3: Ja, laten we het doen dan

P1: Hoewel kosten, dat kan natuurlijk bij heel veel anderen ook zit ik over na te denken. Het is wel een soort heel veel dingen, die CPU kan je ook zeggen, hoe duur is het

P2: Kosten is ook wel iets wat je in iteraties kan vergelijken, is een beetje lastig met iteraties enzo, ga je dan in de volgende iteratie bekijken hoeveel het dan zou kosten

P1: Ja dat is wel inderdaad, dat maakt het ook wel lastig ja

P3: Nou ik denk dat je het liefst zo min mogelijk aan euro's ophangt, want de euro factuur je natuurlijk ook. Als je dan een jaar na een jaar wilt vergelijken en je hebt *inaudible*. Dus ik denk dat je zo min mogelijk aan euro's wilt ophangen, alleen als het het beste is

P2: Ja, als ineens het uurloon is omhoog gegaan dan heb je al

P3: Precies, dan moet je ook inflatie meerekenen

R: Metric kosten kan ook een hele andere unit hebben, zoals bijvoorbeeld FTE's. Hij staat nu in kosten

P4: Maar manuren is ook goed

R: Kosten kan natuurlijk verschillende eenheden aan vast

P3: Ja, zeker

R: Zijn jullie klaar om door te gaan naar de volgende? Ga ik even checken hoe we zitten met het tijd constraint. Ik denk dat het perfect uitkomt. Oke, 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

P2: Wat waren de drie woorden ookalweer?

R: Effectiveness, efficiency, en satisfaction. Dus in hoeverre kunnen gebruikers hun doelen bereiken met die drie

P4: Dat is objectief

R: Goed, ik denk dat hier heel erg het voorbeeld aansluit wat ik eerder noemde. Dat een user noemde dat die een systeem heel gebruikersvriendelijk vindt, maar vervolgens veel langer dan de verwachte tijd over een bepaalde taak doet. Dat is het verschil hier tussen een subjectieve en objectieve measure. Dus hoe kan je satisfaction als actie worden uitgedrukt, is hier bijvoorbeeld de vraag.

P2: Dan kunnen we weer naar die eerste terug gaan, en het ook weer heel concreet beschrijven. Aantal mouse clicks

R: Aantal clicks staat nu bij functional suitability, zouden jullie hem daar dan daaronder houden?

P2: Ja, ik wilde voorstellen dat we dezelfde aanpak hiervoor zouden doen zegmaar. Dus al die eisen van usability heel concreet beschrijven zodat het wel meetbaar wordt

P3: Ja, dan heb je gelijk een hele goede. Er is ook nog iets nieuws, en dat is misschien dat je het vergelijkt met degene die het gemaakt heeft. Dus een developer die door een systeem klikt, en dat vergelijkt met een aantal users. En van die users dan het gemiddelde. Dus gewoon je N users. Dat verschil in tijd om een vastgestelde taak

R: Dus developer effort als baseline

P3: Ja, je vraagt een developer om een taak uit te voeren in een systeem. Stel je hebt een Magister, leerlingcijfer systeem. En je vraagt een developer om een taak uit te voeren in het systeem. En je vraagt een docent om een cijfer in te voeren

P1: Maar, waarom dan de developer. Stel een developer maakt twee misclicks, waarom niet gewoon de optimale tijd?

P3: Ja oke, optimale tijd zou ook goed zijn maar hoe zou je optimale tijd doen

P1: Nou, je kan de perfecte tijd niet lukken. Maar stel je hebt een developer, je hebt een andere developer. Hij maakt een misclick, die andere is net iets sneller. Stel je hebt een developer die gewoon een beetje aan het clicken is. Wat doet dat ertoe? Misschien is iemand anders dan sneller dan een developer?

P2: Die kent hem door en door dus dat lijkt mij praktisch gezien wel de meest handige persoon om mee te testen

P3: Of de business consultant die helpt te implementeren

P1: Nee het gaat er niet specifiek om wie dat dan precies doet, maar meer om echt de effort. In plaats van echt de developer, meer de optimale

P3: Maar wat is dan de optimale, is dat een halve seconde keer aantal clicks. En dat is hoeveel seconde je ongeveer

P1: Ah als je in seconden, ik dacht aantal clicks. Ja als je het in tijd doet, ik dacht inderdaad, ja. Of gewoon weer de tijd ten opzichte van de vorige versie

P3: Ja gemiddelde tijd ten opzichte van de vorige versie

P1: Maar een ander ding waar ik ook over na zat te denken. We hebben het over magister maar stel je hebt instagram. Dan wil je helemaal niet dat het zo snel mogelijk gaat. Je wilt niet dat mensen door gaan scrollen

P2: Ja, maar usability van bepaalde stukken, bepaalde dingen zoeken of vinden

P1: Ja, fair enough

P3: Ik denk eigenlijk dat met je, hoe we ook bij de eerste hebben, dat we daar al een heel omvattende mee hebben, en verschillende aspecten smart doelen kunt formuleren

R: Ik denk ook relevant is op wat voor soort aspecten zou dat dan mogelijk zijn. Wat voor aspecten maken het systeem usable. Wat voor doelen zou je jezelf stellen?

P3: Ja, goede vraag

R: Jullie noemden net bijvoorbeeld clicks, tijd

P3: Het is denk ik weer gekoppeld aan een bepaalde functionaliteit, die usable is. Dan kan je zeggen deze functionaliteit heeft een happy flow, hoe je er zo snel komt. Zo van deze click, dan die click dan die click. Dan kan je zeggen het zijn drie clicks, dus dan 12 seconden kan je verwachten. Maar dan moet je voor elk systeem weer los de happy paths definiëren

P1: Conversie, als in hoeveel mensen kopen het product

P3: Conversie, maar niet elke applicatie heeft zoiets. Maar een beetje in die termen wel. Dat je kan zeggen elke functionaliteit begin je op home screen, kom je uit bij functionaliteit. En dan de conversie van home screen naar dat gedaan hebben. Dan kan je dus inderdaad dat als metric hebben

P1: Oke, ik schrijf hem op

P3: En als je dat smart opschrijft. Hoeveel mensen lukt dat binnen anderhalve seconde. Of bij hoeveel mensen lukt het binnen een halve seconde plus aantal clicks

P1: Oke, conversie, tijd, clicks

P4: En in absolute getallen dan, hoeveel mensen überhaupt

P1: Ik dacht dat we het allemaal SMART hadden

P1: Dus zeg maar, we willen tachtig procent conversie

P3: Oh smart, dat we uiteindelijk ja of nee kunnen beantwoorden

P1: Ja, we willen dat een user het in drie clicks doet

P3: Ja, en dan kun je zeggen. Dan verschilt het per functionaliteit. Zou ik bijvoorbeeld geven een halve seconde per click. En je gaat voor een functionaliteit bedenken, in hoeveel clicks kan dit. Dus login, zou je in twee clicks moeten kunnen. Is login gelukt in twee clicks, dus 1 seconde. En dan is het ja of nee. Dus ik denk dat je hem zo SMART kan maken

P1: Dat denk ik ook. Gewoon de conversie van zoveel percentage. Tijd, ja conversie tijd. Tijd, conversie, clicks

R: Dat zijn de drie componenten: effectiveness, efficiency, satisfaction. Dus ik denk dat die ook misschien

P1: Dus misschien hoe vaak mensen terugkeren. Daar heb je een term voor

P4: Ik weet wat je bedoeld

P1: Recurrence?

R: Ik snap wat je bedoelt

P1: Dus stel je voor je verkoopt iets, gaan mensen weer naar jouw website terugkomen

P3: Dat heeft een naam ja

P1: Het terugkeren zeker. Misschien hoeveel euro's ze uitgeven

P3: Ja maar niet elke applicatie heeft daar iets mee te maken

P1: Nee dat is waar

P3: Ik bedoel als je een ERP applicatie binnen een bedrijf hebt

P1: Nee maar dat is het terugkomen natuurlijk ook. Als het je baan is, al vind je het een dikke schijt applicatie dan moet je die cijfers toch in Magister krijgen. Laat ik het zo zeggen, als student had ik niet echt een keuze

P3: Kan je ook zeggen, hoe lang je rondhangt. Of je wilt dat dat zo min mogelijk is

P1: Maar dat is tijd toch? Soms is het goed, soms is een SMART goal dat het veel is, soms dat het weinig is. Ik denk

P3: Ik zit te denken satisfaction maar dat is dus lastig

P4: Ik denk terugkerende mensen dat dat nog steeds

R: Oke, zullen we dan over gaan naar de laatste?

P1: Zeker

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

P2: Ik moet ervandoor, is er nog iets voor de laatste?

R: Is oke, dankjewel <naam participant>

P4: Kan je hem nog een keer herhalen?

R: Yes, security is de degree to which a 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

P1: Zo, dat is lastig man

P3: Dus gewoon jouw software systeem is in zo mate beschermd dat anderen gewoon precies zoveel als ze zouden mogelijk zien, zien. En met anderen wordt dan bedoeld andere software systemen, of andere personen toch?

R: Exact ja

P1: Op een heel globaal level schijnt er wel zo'n ding te zijn als risico keer impact. Maar dat is super vaag. Dat is volgensmij hoe er in de baseline over wordt gesproken. We lopen veel risico. Is de impact laag, dan boeit het niet dat er veel risico is. Is de impact hoog, dan is het echt niet oke dat er risico is

P3: Maar, hoe kan je daar wat concreter van

P1: Geen idee

P4: Het is een ja nee eigenlijk, is het ja nee

R: Het hele idee ook van deze standaard is ook dat er bepaalde degrees van veiligheid kunnen zijn

P1: Maar hoe meet je dat? Volgensmij zelfs Apple kan gehackt worden, als je maar genoeg geld hebt. Maar weet je, dat is natuurlijk

R: Maar daarom zijn al deze characteristics zo gedefinieerd dat ze beginnen met de degree in which puntje puntje. Dus hier is het de degree to which a 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

P1: Maar ik denk ook, in een vage term zou ik het wel kunnen definieren. Dus rondom impact keer risico. Maar naar concrete dingen zou ik dit niet kunnen doen

P3: Je kan wel dingen noemen, dus bijvoorbeeld je database is zonder in te loggen bereikbaar. Maar hoe ga je dat meten zeg maar. Dat is iets wat je uit een PEN test rapport kan krijgen, maar. Heel lastig. Je kan een evolutionary algoritme door je applicatie kan gaan, en kijken in hoeveel tijd die informatie komt die die niet mocht hebben

P1: Als het zo makkelijk was

P3: Nee dat gaat niet gebeuren. Als je een hele slechte applicatie

P1: Nee, als dat inderdaad werkt. Dat je op die moment even de lijpste malware ontwikkelt

P3: Misschien als we kijken naar een auto, wat zijn dan de voorbeelden. Dan kunnen we kijken en vertalen naar makkelijk concreet naar software

P1: <naam researcher>

R: Bij een auto zou je natuurlijk kunnen kijken in hoeverre werkt de airbag dat je verwacht dat die werkt, vorm van security

P3: Ja precies, en dat heb je bij een applicatie dat bepaalde dingen gelogd wordt

P4: Zijn dingen aanwezig

R: Ja, een metric kan natuurlijk ook gewoon een boolean waarde zijn. Is een best practice aanwezig of niet. En als je dan een complete lijst aan best practices hebt, zou je dan stellen dat je een complete benadering hebt

P3: Niet gek denk ik

P1: Ja

P3: Nou heb ik toevallig al een keer onderzoek gedaan naar cybersecurity, maar weet ik helemaal geen best practices

P1: Ja oke, maar dat zijn er denk ik echt heel veel hoor

P4: Die OWASP dingen. Lijst met applicatiesoort, pas hier op, *inaudible*

R: Het zou natuurlijk ook kan zijn dat je security het beste kan benaderen aan de hand van een existing checklist

P1: Door wat voor checklist?

R: Een bestaande checklist. Of door een andere standaard. Dat kan natuurlijk ook

P1: Maar is het idee niet met security dat er ook. Oke je hebt die best practices. Elk serieus bedrijf heeft dat. Misschien ook niet I don't know. Maar daar bovenop heb je dus al die kleine dingentjes. Kleine bugs, kleine foutjes. Waarin hackers hun weg vinden. Dus gedeeltelijk kan je het vatten in een lijst met best practices, maar ik denk dat je daar

P3: Dat gaat eigenlijk om design of error. Design is best practices, dan kan je een score geven hoe goed is je design. En verder fouten, dat kunnen we eigenlijk niet meten. Dus errors waardoor, of van die dingen die je vergeten bent, dat is niet te meten

P1: Volgensmij heb je wel of niet geautomatiseerde test software, van die security, PEN test software. Maar daarbovenop heb je ook altijd het *inaudible*, dus PEN test hackers. Ethische hackers. Maar dat is oprecht daar.

R: Dat is natuurlijk ook een uitkomst. Dat die niet meetbaar is, dat het niet gewenst is

P1: Nou, laten we dan die lijst van best practices

P3: Dan heb je je design wel gemeten

P1: En dan errors kun je niet meten

P3: Dus het werk van een hacker

P1: Misschien ook die auto PEN test software

P3: En dat vergelijk je dan ook weer met baseline vorige versie

P1: Precies ja. Alrighty

R: Oke, dat was hem heren. Super bedankt voor het meedoen aan het onderzoek. Dan ga ik nu de recording afsluiten

P1: Ik ben heel benieuwd man, hoe andere groepjes hier mee om gaan