R: Oké, dan ben ik nu de opname gestart. Allereerst bedankt dat je mee wilt doen aan mijn onderzoek. Ik ga mijn scherm met je delen, zodat het duidelijk is wat we zullen doen. Allereerst kan je mijn scherm zien?
P: Ja, zo klein is mijn scherm.
R: Oké, perfect. Dan maak ik hem een stukje groter. Is het zo leesbaar voor je?
P: Perfect, ja.
R: Oké, top.Ik doe dus onderzoek naar software product-kwaliteit. En daarvoor heb ik gekeken naar een ISO-standaard die in de wetenschap veel besproken is. Dat is de ISO 25010.
En ISO 25010 beschrijft software product kwaliteit aan de hand van acht kwaliteitskarakteristieken. Dus de kwaliteitskarakteristieken zijn abstracte begrippen.
En in dit onderzoek hebben we ons allereerst gericht op deze acht karakteristieken meer concreet maken.Dat is gedaan door discussiegroepen.
Tijdens de discussiegroepen hebben we geprobeerd om kwantitatieve metrics te vinden voor deze acht karakteristieken. Dat hebben we gedaan. Daar is een totale lijst uitgerond.
En het doel van het komende uur, van dit interview, is om de gevonden metrics te vergelijken, allereerst met de source. Dus waar kan je deze metric mogelijk meten?
Waar haal je de data vandaan? Is dat bijvoorbeeld in Jira?Is dat de codebase die je in een repository vindt? Is het een bepaalde dashboard, misschien een tooling die je gebruikt? En vervolgens gaan we ook nog kijken naar twee dimensies. Allereerst de difficulty of obtaining data. Beide dimensies zijn ingeschat op low, moderate of high. Dus als de data eigenlijk met vrij weinig werk, een uur tot een dag, gemiddeld gezien, available gemaakt kan worden, dan schat hem in als low. Duurt het een dag tot een week gemiddeld en effort is nodig om de data beschikbaar te maken, dan schat hem in als moderate. En als er echt extensive effort voor nodig is, dan wordt hij ingeschat als high. En vervolgens hebben we de dimensie technische expertise. Die heeft ook low, moderate, high, waarbij low gewoon basic technical skills zijn om de measure te kunnen implementeren. Bij moderate zijn wel technische skills nodig, overvolende specifieke tools, maar niet in-depth advanced knowledge. En high technical expertise betekent dat je echt hele specifieke in-depth knowledge over een specifieke tool nodig hebt om de measure te kunnen implementeren.
Zijn de verschillende dimensies duidelijk voor je?
P: Redelijk, ja. Laten we met de format beginnen.
R: Jazeker, ja. Ik denk dat het het beste is om te doen. Ook belangrijk om nog even bij te vermelden is dat we voor nu maar naar een aantal van de karakteristieken gaan kijken puur vanwege een tijdsconsern. Yes? Oké. De allereerste is performance efficiency.
En de definitie daarvan in de standaard is the performance of a product or system relative to the amount of resources used under stated conditions. En zoals net ook al gezegd, de gevonden measure daarbij, één van de 17 gevonden measures, is memory utilization.
Dus eigenlijk voor elke measure gaan we nu de volgende drie vragen doen. Gezien jouw werk, jouw werk bij <naam organisatie> en de technieken die jij gebruikt, hoe zou je memory utilization mogelijk kunnen werken? mDus wat is de source van de data? Hoe moeilijk is het om de data te obtenen? En hoeveel technische expertise is required? Yes?
P: Ja.
R: Oké. Dus dan laten we door met de eerste vraag.
Hoe zouden we memory utilization binnen jouw context het beste kunnen meten?
P: Nou, we werken inderdaad serverless. Uitsluitend serverless. Dus we hebben niks wat op virtuele machines draait of in die zin helemaal niet. Ja. Dus er zijn een aantal resources die wij standaard gebruiken.

En dat zijn de standaard computer resources van AWS in dit geval. Dus je hebt de Glue, de Lambda. En wat andere losse resources. Dat is wel hard op te denken.
Nou, laten we met die twee beginnen. Daar zou je gewoon de runtime kunnen berekenen en hoeveel resource units zijn gebruikt. Daar staat je gebruik van. De Glue gaat bijvoorbeeld in DPU. Dat is niet perse RAM of memory. Maar dat is wel representatief voor een bepaalde hoeveelheid memory en CPU die je gebruikt.
R: Ja, dus de service.
P: Dat is goed begrepen. De service zelf, die biedt daar al eigenlijk in een interface metrics voor aan. Zouden we dus op een account hier voor bijvoorbeeld. Zouden we dat soort dingen kunnen monitoren. Waar het nu nog niet gedaan wordt. Daar doen wij niks mee.
R: Deze verschillende metrics waar je het over hebt. Hoe lastig is het om de data te obtainen? Dus de bovenstaande scale. Misschien is het ook zinnig, maar als ik deze even naar je stuur.
P: Ja, dat is goed.
R: Dan kan ik hem daar even bij houden in plaats van dat we terug hoeven te gaan.
P: Even denken hoor. Je vraagt pas hoe moeilijk het is om die data te krijgen. Dat is relatief makkelijk. Twee routes gebruiken. Eén is gewoon een oplossing vanuit onze AWS omgevingen. Waarbij gewoon de CloudWatch, de, hoe heet die? De Cost Explorer, dat soort dingen. Dat soort toolen gaat gebruiken. Dus daar staat die data eigenlijk al in.
Misschien moet je daar nog even een paar records om dat je moet op een *inaudible* ofzo. Om precies de metric te krijgen die je wil. En de AWS moet die daar gewoon oplossingen voor aan.
R: Ja, dus omdat het al standaard aan oplossingen zijn, zouden ze dan onder low of moderate kunnenklassificeren?
P: Nee, low. Zeker low. Dat is heel simpel te doen. En eigenlijk kunnen we hetzelfde ook meteen zeggen voor CPU. Want die gaan in de hand. Dus eigenlijk is dat hetzelfde.
R: Die worden samen aangeboden zeg jij. En aangezien wij bijvoorbeeld geen data warehousing of dat soort dingen doen, maar zelfs dan zou je binnen de cloud omgeving dat gewoon allemaal low effort kunnen ophalen.
R: Ja, dus low. En de technical expertise required?
P: Ja, dan zou je kunnen zeggen low op het moment dat je wel bekend bent met cloud. Dus ik zit even te denken waar zeg maar de basis cloud kennis hoort. Is dat low of is dat moderate al?
R: Ik zou deze persoonlijk denk ik onder low dan scharen. Dus basic cloud kennis.
Waarbij moderate misschien specifieke kennis over een bepaalde service is. En waarbij high zou zijn in depth knowledge over die service.
P: Ja, nee. Je hoeft eigenlijk in feite weinig kennis te hebben. Behalve dat je dus wat ik zei bij Glue heb je bijvoorbeeld wat uitgedrukt in DPU's. Dus dan moet je, dat zou net even wat reken sommetjes die je dan zelf moet bijdenken zeg maar.
R: Ja, ja.
P: Dus dan kunnen we deze ook onder low classificeren zeg jij.
R: Vervolgens hebben we een soortgelijke die ook valt binnen hetzelfde straat als memory and CPU utilization, network utilization.
P: Daar wordt het een beetje vaag he, want dat zit in de cloud. Dus zeker met circular architectuur heb je geen netwerk in eigen beheer. Dat is allemaal gemanagd door AWS. In ons geval door AWS. Dus dat meten wij ook niet.
Kijk, als wij bijvoorbeeld wel virtuele netwerken en dat soort dingen zouden doen in de cloud. Dan zouden we dat kunnen gaan meten en dan zouden we daar ja,

dan haalt dat verkeerd maar dat doen wij niet.
R: Dus je zou zeggen dat network utilization eigenlijk not applicable is binnen jouw context omdat er volledig servers wordt gewerkt. Vervolgens gaan we dan door naar de latency per user. Dus de gemiddelde respons tijd voor een user.
P: Is in die zin van toepassing omdat we daadwerkelijk wel echt gebruikers hebben.
En dan zijn er eigenlijk twee typen gebruikers die daarin herkennen. Dat zijn eindgebruikers die bijvoorbeeld een query uitvoeren op de data. Gebruikers die een AI model trainen. Maar echt de mens zeg maar, een echt mens. En we hebben systeemgebruikers en dat zijn bijvoorbeeld data warehouses die data consumeren en dat zijn meestal de grotere volumes . Daar zou je latency kunnen meten.
R: Oké, waar hou je dan de data vandaan?
P: Ja, dus het is dat je daar zeg maar de sprong moet gaan meten van waar de data staat naar waar die geconsumeerd wordt. En waar die geconsumeerd wordt, dat is een heel divers palet aan oplossingen. Dus daar is niet één oplossing die dat kan doen voor je. Dus stel we zouden dat, bij een reguliere webapplicatie ofzo is dat heel makkelijk. Want dan meet je gewoon de server site en de client site zeg maar aan de twee kanten
en dan heb je je latency bedacht. Maar in dit geval is er dus, de client site is heel divers en heel complex. Dat kunnen wel ineens data warehouses zijn of andere systemen. Met APIs Dus ik zou zeggen dat de bron is heel divers en dat maakt het ophalen van die data ook heel lastig omdat je heel veel verschillende, je zou voor één soort systeem zou je dat kunnen gaan meten bijvoorbeeld.
R: Maar hoe nuttig is dat?
P: Dus je zou eigenlijk zeggen, de source kan hier heel divers zijn omdat je eigenlijk heel veel verschillende smaken van consumers hebt. Dus afnemen in applicaties en eigenlijk latency anders gemeten kan worden per applicatie.
R: Hoe zou je dan de difficulty of obtaining data inschatten?
P: Ja, ik zou zeggen voor een enkele bron is dat nog wel goed te doen. Dus voor bijvoorbeeld, denk bijvoorbeeld even aan CBS. Die hebben natuurlijk een paar mooie afnemende systemen liggen. Daar zou je redelijk, ik zou zeggen moderate voor een enkele, maar op het moment dat je het voor allemaal moet doen, voor al onze afnemers, en lateral hebben we veel afnemers. Echt veel. Binnen heel <naam organisatie>. Iedereen neemt data van datalake af. Daar zou het best wel eens zeg maar. Dan wordt het echt high zeg maar. Dan wordt het high complexity.
R: Oké. High difficulty.
P: Even een andere vraag aan jou. We hebben natuurlijk van datalake beschrijven. In feite zouden we als datalake een doorgeeft luiken. Zo moet je het een beetje zien. Dat is natuurlijk wel iets complexer dan dat. Maar uiteindelijk is een afnemer voor ons iemand die geïnteresseerd is in de bron data. En wij zitten er als pipeline daar tussen. Nou zouden we bijvoorbeeld latency, zouden wij ook kunnen beschrijven in hoeveel tijd het ons kost om die data van de bron naar de afnemer te brengen. Maar dat is volgens mij niet wat je hier bedoelt, toch?
R: Nee, dat is inderdaad niet wat je hier bedoelt. Dit is meer van de consumer die een request naar jou stuurt. Wat is de response tijd daarvan?
Dus inderdaad aan de andere kant. Maar ik snap je punt inderdaad. Alleen dat is niet waar we hier naar kijken.
P: Nee, dus dat zou dan difficulty high nog steeds. En technical expertise ook high.
Omdat je een diverse portfolio hebt. Dus je moet veel verschillende technologieën kennen. Veel verschillende systemen. En er is niet één oplossing voor.

R: Ja, exact.
P: Dus het is niet makkelijk.
R: Als ik goed een beetje een summary maak van wat je zegt. Het is niet makkelijk om dit te doen vanwege de hoge diversiteit. En daar dan moet je dus over verschillende technologieën, in-depth knowledge hebben om dit te kunnen implementeren.
P: Ja.
R: Als ik dat zo hoor, betekent dat ook hetzelfde kunnen invullen voor de volgende. Dat is namelijk de latency per increasing user. Die is niet helemaal self-explanatory. Dus vandaar dat hier een extra uitleg bij staat. En in de discussiegroep hebben de deelnemers gezegd, dat de latency per increasing user, bedoelen ze dus, wanneer je het aantal gebruikers van x naar y increased, wat doet dit met de gemiddelde latency? Dus als de vorige al diverse high is, kunnen we dan hetzelfde hiervoor invullen?
P: Nou, daar kunnen we wel iets anders bij bedenken. Aangezien wij serverless gaan, zou een toename in gebruik vrijwel geen effect moeten hebben op een verschil in latency. Dat we de latency moeilijk kunnen meten. Wat we wel weten vanuit de theorie, zeg maar, is dat
R: Ja, vanuit de theorie zou het geen verschil moeten hebben,
maar om daadwerkelijk die getallen te krijgen,
of het geen verschil is, dat is waar we hier geïnteresseerd in zijn.
P: Uiteindelijk zou je in case-by-case zoiets moeten gaan bekijken. High, high.
R: Nog steeds, oke. Goed. Nou, de volgende gaat al wederom over latency.
Dus wellicht kunnen we hetzelfde invullen, alleen kunnen we nog even kort bespreken. Dat is latency difference per location. En met een locatie bedoelt men hier echt een geografische locatie. Dus iemand die jouw service vanaf… Vanaf Europa-request versus iemand die jouw service vanaf… Misschien Zuid-Amerika-request. En dat gaat dan specifiek inderdaad over hoe kunnen we dit meten en hoe moeilijk is het om deze getallen te obtainen.
P: Uiteindelijk dus hetzelfde als überhaupt latency meten.Maar als je dat één keer kan, dan is het niet zo moeilijk meer. Dus het is een beetje… Maar pure latency meten is complex. Latency meten is difficult. Maar als je eenmaal latency meet, zeg jij dat de extra effort om zowel een vorige te krijgen als deze, dan is dat niet heel veel extra effort.
Nee. Nee, want uiteindelijk is dat dus… En hoe zeg je dat? Dat is het voordeel aan latency meten op die manier. Dan heb je hem al gesegmenteerd liggen, zeg maar. Dat doe je automatisch op afnemen. Dus een locatie of een toename in gebruik of al dat soort dingen. Dat zou je allemaal zichtbaar moeten hebben al.
R: Ja, dat zijn dimensies over de vorige. Ja, dan zet ik hem voor nu zo. En dan zet ik even een note onder: “Considering latency measurement is in place.”Zet ik die hier even als extra dimensies onder.Vervolgens de volgende measure. Het is overigens belangrijk dat dit inderdaad allemaal measures zijn tijdens de focus groups zijn besproken.Dus het is niet belangrijk of je ermee eens bent of niet. Ik heb ook in een vorige interview gehad iemand die zei, nou eigenlijk vind ik het niet een hele goede measure, alleen dat is niet iets wat we nu bespreken, het is gewoon iets wat tijdens interviews is gezegd en dat zijn de kosten van je resources gedeeld door de expected costs. Dus runt je product eigenlijk volgens de kosten die je verwacht? Of heb je vanuit eigenlijk een onverwacht perspectief veel meer kosten gedraaid?
P: Even denken hoor, want dit is in het straatje van hoe moeilijk het is om die data te verkrijgen. Nou, de resource costs, die kan je, dat is gewoon AWS eigen, of de cloud eigen.

Dus je hebt gewoon je cost explorer, je budgeting, dat zit daar gewoon in al, dat hebben we eigenlijk nu al. The expected cost is wat anders. Ik weet niet of wij überhaupt een verwachting kunnen uiten van hoeveel iets gaat kosten. En dat zit vooral in dat die data, groot volume data, 100 data sets, verschillende soorten die allemaal verschillende intervallen hebben waar binnen ze binnenkomen, bepaalde seizoenswerking zit daarin. Tijdens de kerst uit de nieuwe periode, daar hebben we heel veel hogere volume data, daar hebben wij geen verwachting van, wat dat extra kost voor ons. Als we die verwachting wel al zouden hebben, dan is het meten van dat verschil echt heel simpel.
R: Maar het maken van die verwachting kan complex zijn, en als ik je zo hoor, ook niet altijd een known.
P: Nee, precies. En vooral ook, want in ons proces zit natuurlijk een paar karakteristieken. Enerzijds is het, noem het 5 phases van big data, de snelheid waarmee het komt, hoe divers is de data, hoeveel verwerking is het überhaupt. Maar dan vervolgens krijg je eigenlijk, dat is wel interessant om heel even kort over te hebben, je hebt zeg maar, bij ons ligt de transitie van source-aligned data naar consumer-aligned of target-aligned data.
En dat betekent dat wij zeg maar ergens onderweg in dat proces, zeggen wij, oké, dit volgt, de data is allemaal in vorm, zoals hij voor de bron zou moeten zijn. Vervolgens maken wij een versie die goed te exploreren is, zodat mensen er goed naar kunnen kijken, en dan vervolgens krijg je dat er een afnemend systeem, denk een data warehouse, of inderdaad een AI-model, die daar ook moet gaan aansluiten, die heeft ook een bepaalde requirement. Dan moeten we data gaan transformeren in een ander vormaat, of ondanks dat we dus inhoudelijk niet een verandering doen aan de data, maar de vorm van de data moeten we wel veranderen. En dat weten, zeg maar, de impact daarvan kan je eigenlijk van tevoren niet weten. Nee, misschien gewoon low for research cost, not available for expected cost, dat je zegt van, eigenlijk kan je die data niet echt verkrijgen. Ja, dat is de vraag, zeg maar, dan weet ik niet of dat kan. Want dan zou je eigenlijk, wat je dan moet doen, dit is eigenlijk een business-vraag, zeg maar, want technisch gezien kan het, zeg maar. Ja, technisch gezien kan het, maar het verandert wel over de tijd, en het verandert wel over een aantal assen, zeg maar. Dus ik zou zeggen dat de expected cost überhaupt bepalen is een business-proces wat heel high effort is, zeg maar, wat heel veel moeizame, het kan wel, denk ik. Exact, ja. En dan is het nog maar de vraag of het gewend is om het weer nou ook te doen.
R: Oké, en als de technische expertise required, en laten we dan puur ingaan op de resource cost, je zegt dat hij is included in cloud services,
heb je dan ook weinig technische expertise nodig?
P: Ja, kan je echt, als je in kan loggen in een cloud portal, dan kan je de cost gaan evalueren.
R: Laten we dan doorgaan naar de cyclomatische complexiteit. En in die je niet bekend bent, is dat het aantal paden dat je hebt door een software systeem.
Stel je hebt een product, hoeveel verschillende cases kan jouw code doorlopen?
P: En wat bedoel je met een case?
R: Bijvoorbeeld, stel je hebt een if-else, dan zou je een cyclomatische complexiteit van twee hebben, want je hebt twee verschillende paden door je systeem.
Je gaat het ene pad, de if door, of de else. Dat zijn nou twee paden.
En stel dat er bijvoorbeeld weer een ifje in je if is genest, dan kan je een derde pad,
en als daar ook weer een andere conditie aan zit, heb je een vierde pad.
Dus dat wordt bedoeld met het aantal paden.
P: Ja, ik vind het even lastig te vertalen naar wat wij doen,

want het gaat uit een app of zo, waar je naar front-end zit te klikken, dan snap ik het.
Maar in het datalink is het natuurlijk de consument, die wil een dataset zien.
En om die dataset te produceren, is er dan nog een proces. En is dan de collectie tijd, dat proces, hoeveel verschillende verpakkingen we binnen dat proces hebben.
Precies. Dat is precies wat de cyclomatic complexiteit hier is.
R: Ja, klopt.
P: Het ophouden van die complexiteit, zou ik niet per se heel ingewikkeld vinden.
Oké. Even goed nadenken, want we gaan, en dat zit hem vooral in onze nieuwe structuur. De oude structuur is iets complexer, maar in de nieuwe structuur, even uitgaande van de gewenste oplossing daarin, bijvoorbeeld dat wij distributiedata, die splitsen wij op bucket-niveau. Dat betekent dat elke afnemende case, die heeft eigenlijk zijn eigen versie van een distributieset van een dataset. Dat zou je prima dus gewoon, je telt eigenlijk gewoon een dataset, en dan weet je hoeveel verschillende verpakkingen van een dataset er zijn. Daar binnen, binnen een dataset, een distributieset, kan er natuurlijk ook nog wel wat variatie ontstaan. Als er een gemengde structuur en een platte structuur met elkaar naast elkaar geplaatst worden. Dat ligt allemaal onder de doek. Opslag is goedkoop en computer is duur. Dus dan ga je double opslaan, want anders moet je het twee keer berekenen.
R: Exact, ja.
P: Dus daar binnen kan er ook nog wel wat variatie zitten, en dat is echt een case-by-case onderzoek. Dan moet je echt kijken van, hoe is die dataset gestructureerd? Dus ik zou zeggen dat, om die data te verzamelen is niet per se low, maar het is wat we doen. Het is niet high. Nee, dus eerder categorie moderate.
R: Ja, precies.
P: Dus ik zou zeggen, het is mogelijk beschikbaar. Je moet wel even wat werk ervoor verrichten om het goed te krijgen. En inderdaad, als je hier zegt een dag of een week, dat zie ik alvorens.
R: En de technische expertise bij deze?
P: Je moet inhoudelijk weten hoe ons data lake werkt. En dat is een unieke oplossing, zou je zeggen. Het is niet zo bijzonder, maar ik zou hier ook moderate zeggen, omdat je niet zomaar even wat basis cloud kennis kan hebben en dan dat kan doen.
Je moet echt begrijpen hoe ons data lake werkt.
R: En waarom zou je dan voor moderate in plaats van high gaan?
P: Ja, doordat het ook alleen maar een data lake is. Het is niet een AI-model met allerlei black boxes en dingen. Het is alleen maar een dataproces. Dat is wel redelijk goed te krijgen.
R: Ja, oké, makes sense. Doorgaan naar de volgende?
P: Ja
R: Hierbij zegt men eigenlijk iemand stelde voor, je kan performance efficiency meten door te kijken naar the used current. dus gebruik de elektriciteit door je dat door je systeem, gedeeld door de verwachting daarvan. Is dat iets wat überhaupt te spraken is in een data lake solution?
P: Stroomverbruik.
R: Exact, ja.
P: Stroomverbruik. Ja, dat hebben we al cloud-managed. Dus daar hebben we al een paar dingen. Dat hebben we al cloud-managed. Dus daar hebben we niks mee te maken.
Ja, dus in dit geval hebben we geen...
R: Not applicable?
P: Ja, want dat server is allemaal. We hebben geen. Dat kan je niet makkelijk invullen.

R: En bij mobiele apps, de mobile app battery usage.
P: Ja, hebben we ook niet.
R: Dus dat is not applicable.
P: Exact.
R: En als ik jou kees hoor, heb je deze ook niet. Dat is namelijk the time to interactive. En dat is de tijd die het duurt. Tot een webpagina volledig actief wordt.
P: Ja, de webpagina hebben wij niet. Dus dat bieden wij niet aan in dienst.
Je zou het kunnen bekijken of.. Nou, ja, nee. Ik zou niet zeggen dat het applicable is.
Dat is gewoon een andere dienst. Of misschien als de output een dashboard is...
Dan kan dat natuurlijk ook. Dus de tijd totdat een interface interactief is.
Zo zou je hem ook kunnen betrekken.Maar ik weet niet of jullie dat hebben. Er is ook geen dashboard.Dat is allemaal als self-service. Dus dat doen wij niet.Dan kan je zou je kunnen zeggen: er zijn gebruikers die dus een dataset inladen in een dashboard. Dan stap je daarvoor, zeg maar. Maar dat kunnen wij ook niet.
R: Dus deze is not applicable in jouw case.
P: Ja, dat is ook uitbesteed aan de cloud.
R: De volgende is de cost per retry. En in de focus is die geschreven binnen een event-driven architecture. Maar eigenlijk als je. Wat men hiermee bedoelde is. Als je een functie hebt en je moet voor some reason de functie nog een keer runnen omdat die gefaald is. Wat kost je dat? Kunnen wij dat meten? Laten we daarmee beginnen. Kan je het meten? Hoe kan je het meten? Hoe moeilijk is het te meten? En technisch verlaten, wat moet je daar voor allemaal doen?
P: Ons context is dan dat er bijvoorbeeld een dataload binnenkomt. En die veroorzaakt een fout. En dan moet er een handmatige retry doen. Of we moeten daadwerkelijk een systeem afvuren opnieuw. Historical loads opnieuw. Dat zou een redesign vereisen. Dus dan moeten wij technische implementatie doen.Van het verschil tussen verschillende runs.
Dus denk aan een Glue function die afgevuurd wordt. Die nog een keer afgevuurd wordt. Want nu in de meting, zoals je koste zou meten, dan zou je geen verschil zien tussen een reguliere en een retry. Dus dan zou je bijvoorbeeld runs moeten gaan taggen. Op een of andere manier. Dat zou kunnen, maar dat is wel, voor de meeste services is dat nog wel te doen.Maar goed, dat zit dan nog even low effort in de metingen. Ik zou zeggen moderate. Een weekje deven, dan kom je nu wel uit, denk ik. Om die data te kunnen gaan verzamelen. En dan zou ik zeggen expertise high. Omdat de manier waarop wij datalake deployen is natuurlijk best wel geavanceerd. We hebben natuurlijk CDK deployment. Binnen CDK definiëren wij constructs. Constructs zijn niet alleen de CDK constructs, maar ook echt...
De building blocks die wij maken. Die bestaan uit verschillende CDK constructs.
Vervolgens heb je dan een stukje inzicht. Waarbij er dus engineers van buiten het team op ons platform werken. Dus dan zouden wij hun constructs en gebruik van data constructs ook moeten aanpassen. Dus als je zegt om dat goed toe te passen… De technical know-how, die moet wel echt te houden zijn.
P: Ja, begrijpelijk. Oké, en dan eentje die te maken heeft met de vorige. Alleen wellicht nog een stap daarvoor is. Gewoon het aantal retries dat je moet doen.
P: Ja, dat is echt hetzelfde. We gaan nu dat soort van vingerwerk vol kunnen doen. Maar om het echt exact te meten is het hetzelfde.Je moet de runs taggen met of te reguleren op een retry.
R: Ja, ja, makes sense. Oké, de volgende die gepropost is. Is het percentage van dead code. En hierin heeft degene die deze voorgesteld heeft dead code beschreven als de hoeveelheid code waarvan de resultaten nooit gebruikt wordt.

P: Ja, dus dat kan ik op twee manieren interpreteren. We hebben een aantal repositories waar code in zit. En die code hoort nooit afgevuurd. Of die zitten gewoon functioneel, die worden niet gebruikt. Of we hebben bijvoorbeeld data sets die niet gebruikt worden.
Oude legacy data sets waarvan de code dus wel draait. Maar je hebt er geen transmit voor. Maar we hebben geen afnemer.
R: In dit geval bedoelen we de eerste case die je hebt beschreven.
P: Daar heb ik een heel duidelijk antwoord op. Daar zijn we nu bezig om een implementatie van te maken, al namelijk. En dat is via Sonar Cloud, heet dat volgens mij.
Oké. Sonar Cloud is een tool die <naam organisatie> inkoopt. Daar hebben ze een license voor. En die kunnen wij nu integreren in onze GitHub-omgeving. En die doet een analyse van een heleboel verschillende aspecten. Dus een heleboel metrics van ons code metie. Waaronder dode code is een van de aspecten. En dan moet je wel zelf wat definiëren hoe je dat dan precies wil. En dat valt er wel onderweg niet. Dat kunnen wij dus eigenlijk relatief simpel. Ik zou zeggen, low effort om te obtainen. Low expertise kunnen we dat doen omdat het al geïntegreerd zit in onze GitHub-omgeving.
R: Exact, ja. En de technische expertise om deze te implementeren. Heb je dan een specifieke kennis van Sonar Cloud nodig om dit te doen? Of hoe werkt dat?
P: Dus ja, dat heb je nodig. Ik weet ook niet zo heel goed hoe dat in elkaar streekt.
Dat is eigenlijk de teamleder die er nu mee bezig is. Het schijnt wel, als je een keer een beetje een setup hebt, dan is het een redelijk goed te begrijpen tool. Het gaat natuurlijk bij ons wel over. Ja, het valt een beetje samen met je kennis van code überhaupt.
Dat je dit een beetje redelijk kan bevelen of niet. We hebben Python en TypeScript, voornamelijk. Ik geloof dat er nog een klein stukje hier en daar JavaScript is.
Voornamelijk Python en TypeScript. En dan, zolang je die twee zijn, maar niet het low expertise verder. Want de tool wijst zichzelf wel, zeg maar.
R: Exact, ja. Vervolgens heeft men gezegd de number of connections between a source system and other systems. Dus hierin bracht eigenlijk een deelnemer het argument van als je dependent bent om andere informatie op te halen uit andere systemen. Dan performt je systemen over het algemeen slechter omdat je die dependency hebt. Dus daarbij argumenten het number of connections. Dus met hoeveel andere systemen is jouw systeem verbonden?
P: Ja, dat zijn er bij ons veel, weet ik. Is dat makkelijk te meten, ja of nee?
Kan je die tellen, inderdaad? Dat is dan de bijkomende vraag. Ja, het punt is, het zijn er veel. Dus het zijn er goed honderd, zoveel bron-systemen die bronnen die tegen ons aangehouden worden. Sommige komen uit eenzelfde systeem, sommige komen uit verschillende systemen. Dat zou je, je zou letterlijk met de hand 1, 2, 3 kunnen tellen.
Dat kan. Dat is heel inefficiënt en kost heel veel effort, maar het is niet ingewikkeld.
Als je het geautomatiseerd door zou willen meten, dan zou je echt wel iets complexers moeten doen.
R: Exact, ja.
P: Want wat wij doen is wij, we hebben, voor ingest hebben wij gewoon een aantal buckets. Dus we hebben, ik ga er even uit van onze nieuwe situatie. Dus de Legacy V1 die jij bijvoorbeeld ook al kent van de Data Lake Team. Ja, zeker. Dit gaat even over de V2 variant, zeg maar. Ja. En daar definiëren we dus per data set een bucket.
Zeg ik dat goed? Ja. Nee, dat klopt. Dus per data set een bucket. Dat betekent dat je eigenlijk die buckets zou kunnen tellen. En dan even moet kijken welke er zeg maar uit eenzelfde systeem komen.
R: Exact, ja. Dus als je het aantal connecties wilt tellen, dat is even, het is niet ingewikkeld.

P: Het is gewoon veel, zeg maar. Ja, veel werk. Dus dan in dit geval veel werk, maar weinig difficulty. Is dus weinig technische expertise, maar wel niet zoveel difficulty of opdeling data omdat het wat tijd kost? Ja, ja, ik zou eigenlijk nog wel een low zetten eigenlijk.
Want als ik gewoon twee uur achter elkaar zo met mijn vinger op het scherm ga tellen, dan heb ik antwoord. Ja, dat is voor de difficulty of opdeling data.
R: Voor welke van de twee dimensies heb je het nu over?
Allebei, want het is, ja. Deze kun je tellen, zeg maar. En je kan het zo een keer weer maken als je zelf wilt natuurlijk. Je kan dan even geautomatiseerd het systeem en bedenken, maar het hoeft niet.
R: Exact, ja. Goed, vervolgens de ene laatste, de lead time van een functie. Dus hoe lang duurt het om een bepaalde functie uit te voeren?
P: Ja, en een functie in mijn context is dan een ETL proces, misschien?
R: Exact, ja. Dus je noemde een glue als gebruikte service, dus hoe lang een glue job runt. Of hoe lang een laptop functie runt.
P: Ja, dat is prima te meten, want je kan gewoon jobruns. Je kan gewoon, als je het gemiddelde van een jobrun wil meten, dat is prima te doen. Dus dat is redelijk low effort. Als je een basisbegrijp van glue hebt, dan kan je dat prima ophalen. We hebben natuurlijk wel nog wat iets we hebben die event bridges en zo ertussen zitten. We hebben wat notebooks. Uiteindelijk, als je iets van basisbegrijp van zo'n service hebt, dan kan je daar wel naar uitvoeren, zeg maar.
R: Exact, ja.
P: Dus vandaar dat we dat gewoon weer dit kunnen invullen.
R: De volgende measure is eigenlijk dezelfde, alleen dan tegen een iets complexere wellicht tegenhouden. Dus je meet eerst de lead time en vervolgens kom je erachter dat de lead time eigenlijk niet gewenst is. En je gaat een deel van de code refactoren. En vervolgens meet je weer de lead time. Alleen je bent juist geïnteresseerd als output het verschil tussen die twee runs.
P: Uiteindelijk is dat niet heel complex, want je kan het gewoon, low effort kan je, ik weet het wel, ophalen, zeg maar. Ook als je een wijziging doorvoert, ook in je doet dan run.
Het enige wat bij ons lastiger wordt, is dat die run times die variëren natuurlijk heel erg op basis van de invoerde data. Dus dan zou je met, als je dit zei dan weer uit een testbegeving, zou je dit doen of zo met een identieke set, een aantal run doen. En dan, ja, dan is dat redelijk makkelijk te doen. Dus dan schat hij ook weer zo in.
R: Oké, dan hebben we nu de 17 voor performance efficiency gedaan. En omdat dit ook het, wat ik aan het eind van elk interview doe, is ook gewoon nog een korte evaluatie.
Daarin stel ik een paar vragen. Allereerst de vraag, waren de dimensies duidelijk voor? Heb je het gevoel dat er een dimensie mist? Heb je het gevoel dat de dimensies duidelijk zijn?
P: Nee, ze zijn heel duidelijk. Er staat zelfs een tijdsindicatie bij, dus dat is wel prima.
Oké. Als je het over de technische expertise hebt, is het iets complexer omdat expertise kan natuurlijk op een heleboel aspen zitten. Dus als je zegt, ik ben een Python developer, dan heb je keurig netjes je PC, EP en je certificatieniveaus en zo. Dan kan je zeggen, nou, dat zitten bepaalde expertise erbij. Bij cloud infrastructure heb je weer een hele andere set met expertise.
Ja. Snap je, dus ik kan me voorstellen dat als je veel verschillende interviews doet met een softwarebedrijf, een frontender, een backender, een data engineer, dat het verhaal van technische expertise nogal varieert.

R: Ja, ja. Dat kan ik begrijpen inderdaad. Oké, goed voor mij in ieder geval om over de volgende interviews te nadenken. En dan wellicht wat een idee kan zijn is om aan het begin dan nog iets meer een kader van het werk te schetsen. Zodat ik diegene aan de hand kan nemen wat low, moderate en high qua technische expertise is. En dan eigenlijk de laatste vraag die ik wil bespreken is over het interviewprotocol. Wat vond je ervan?
Wat zijn je ervaringen om dit, om samen met mij deze sessie te doen? Waren bepaalde dingen onduidelijk? Zijn er dingen die ik nog beter kan doen in het volgende interview?
P: Even denken hoor. Dat is een hele brede open vraag.
Ja, zeker. Nou, wat ik heel leuk vind is dat dit onderwerp best wel gewoon slaat op data-gedrevenheid. Ik heb toevallig een teamgenoot die heel graag data-gedreven gaat werken in het data lake. En dan denk je, ja, nogal wie is het in het data lake?
Maar ik bedoel, ons werk data-gedreven maken.
R: Dat is inderdaad ook de hele bedoeling van deze studie. Dat als je iets wil zeggen over de performance efficiency van een bepaald product. Als we onze definitie van memory utilization pakken. We weten hoe we die kunnen meten. We kunnen nu aantonen dat dit product betere performance efficiency heeft dan hetzelfde product in een vorige iteratie bijvoorbeeld. Dat is de idee voor het interviewprotocol.
P: En dat biedt wel ook weer wat inspiratie. Ik denk dat het bijvoorbeeld voor jou ook wel interessant zou zijn. Dat is Kevin. Misschien ken je Kevin al? Van ons team.
R: Zwartsgronden?
P: Ja, zeker.
Ik denk dat hij een hele interessante is om dit interview ook een keer mee te doen.
R: Ja, ik ga hem even een berichtje sturen.
P: Het is natuurlijk hetzelfde team, hetzelfde product.
Voor hem zou het heel interessant zijn voor ter inspiratie ook dit. Het lijkt me ook leuk om dit soort stukken mee te nemen in het volgende portaal. Om daar bijvoorbeeld wat meer werk aan ons kan voor te gaan verrichten. Dat we dit ook echt gaan doen.
R: Ja, exact. Nou, thanks. Dus dingen die ik nog beter kan doen tijdens dit interview?
P: Nee, het meeste is besproken. Inderdaad, wat je zei, dan bevinden we iets meer de toepassingen op de individuele case uitlichten.
R: Oke, top. Goed, dan ga ik de recording afsluiten. En dan wil ik je in ieder geval enorm bedanken.