

Een requirements engineer in een DevOps omgeving

Als requirements engineer moet je met je tijd mee gaan; wat gebeurt er in de ICT-wereld en hoe beïnvloedt dat het werk? De laatste jaren zie je dat de ICT zich vooral is gaan richten op Agile & Scrum werken. Nu is er een nieuwe beweging in gang gezet die ook onze aandacht vraagt. Ik heb het hier over DevOps. Sommige bedrijven doen al DevOps, en veel andere bedrijven zijn aan het bedenken of ze er ook iets mee willen. Zelf ben ik van oorsprong requirements engineer, en ben momenteel ingezet als Scrummaster in een DevOps team. Ik zie dat mijn requirements collega's steeds vaker in DevOps omgevingen worden ingezet. Het is dus hoog tijd om te kijken wat DevOps voor requirements engineers betekent.

door Linda Haak - van der Spek

Wat is DevOps?

Om te beginnen is het belangrijk om te weten wat DevOps is. Dit is ook gelijk een lastig punt, DevOps is namelijk geen methode, maar een mindset. Er is geen prachtig manifest zoals van Agile, of een mooi handboek zoals voor Scrum. Nee op internet is er eigenlijk weinig concreets over te vinden. Duidelijk is wel dat DevOps de combinatie is van Dev (development) en Ops (operations).

DevOps is geen methode, maar een mindset. Er is geen prachtig manifest zoals van Agile, of een mooi handboek zoals voor Scrum.

Wat ik onder DevOps versta is het volgende: DevOps vloeit voort uit Agile. Waar we met Agile zijn begonnen met kort-cyclisch te werken, met business en development bij elkaar aan een product te werken, met korte communicatielijnen, wordt dat met DevOps uitgebreid door aan de Scrumteams de OPS mensen, de operations mensen (functioneel beheer, technisch beheer, database administrators etc.), toe te voegen. Dit met als doel om continu (en daarmee dus nog efficiënter) werkende software te leveren. Dus eigenlijk maken we de stap van Agile/Scrum in Dev-teams naar nog meer samenwerken in DevOps-teams. Om een DevOps-team te worden zijn er allerlei hulpmiddelen beschikbaar die je

daarbij kunnen helpen zoals automatisch testen en deployen.

Bij DevOps wordt naar de gehele IT cyclus gekeken, dat zie je in de afbeelding hiernaast. We zijn begonnen met de stappen specificeren t/m testen te optimaliseren met behulp van Agile/Scrum. Daarna hebben we hier accepteren t/m run aan toegevoegd, bekend als Agile Beheer. En nu maken we de cirkel rond, en nemen dus alle aspecten mee. Dat noemen we DevOps.

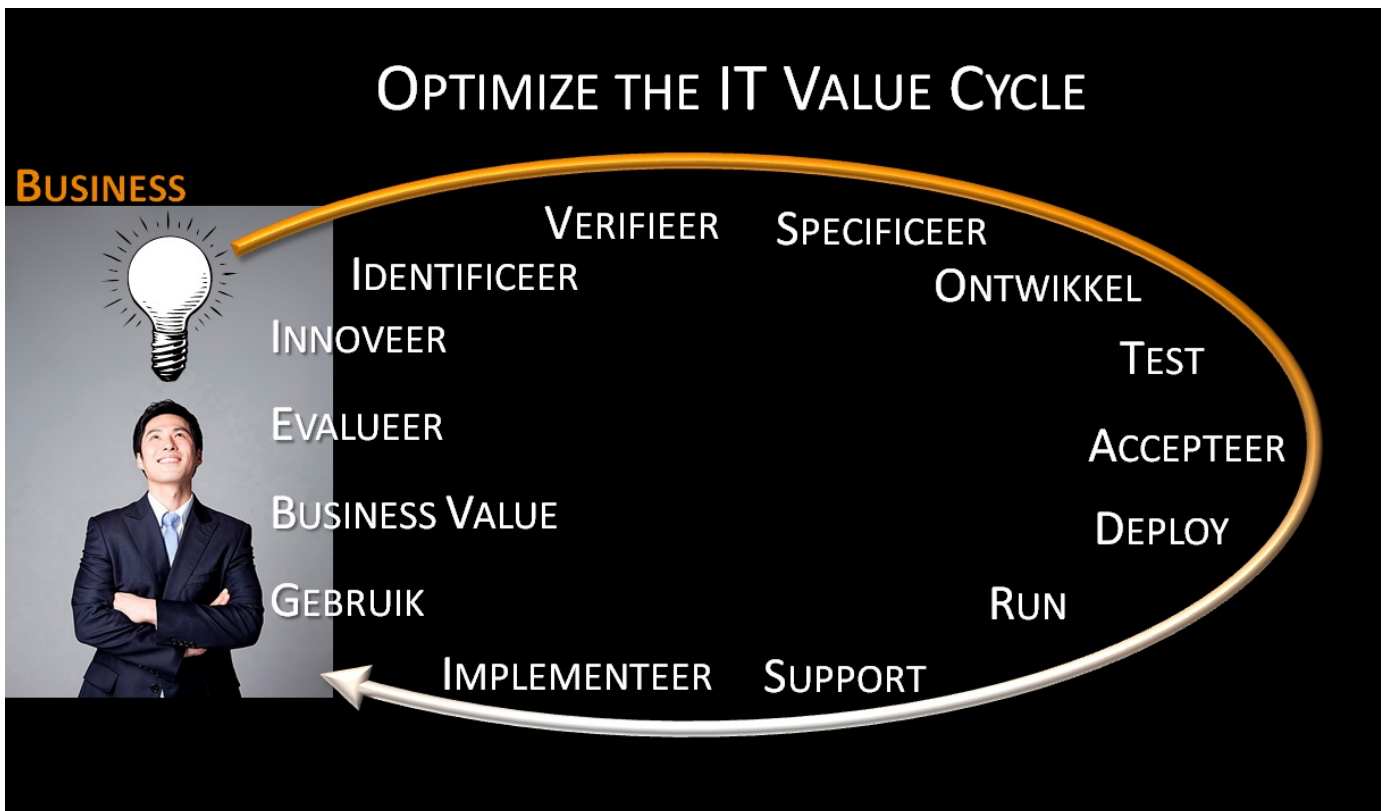
Met DevOps bekijk je elke processtap en bepaal je waar verbeterpunten zitten waar de business profijt van heeft. Daar ga je mee aan de slag. Dit resulteert in een unieke invulling van DevOps in elke organisatie.

Wat zijn de gevolgen voor een requirements engineer?

DevOps is een logisch vervolg op Agile, daarom vergelijk ik de rol van een requirements engineer in een DevOps omgeving met die in een Agile omgeving. Ik zie hier een aantal belangrijke veranderingen.

Van projectlid naar lijnfunctie

Een requirements engineer is altijd gewend geweest om in projecten te werken. Met DevOps verandert dit, je komt namelijk in een lijnfunctie terecht. Het doel van je werk verandert daar deels mee, want ging je eerst naar een volgende opdracht als het systeem 'af' was, kun je nu zomaar jaren blijven werken aan een systeem (jarenlang één opdracht, iets waar we met Agile werken vanaf leken te zijn).



Het betekent o.a. dat je soms wat meer bezig zult zijn met het verbeteren van processen om goed werkende software op te leveren, omdat je hier lange tijd baat bij hebt. Denk hierbij aan bepaalde belanghebbenden die je met regelmaat nodig hebt, of aan het gehele voortbrengingsproces.

Wat ik interessant vind is bijvoorbeeld 'dark launching' en 'levers & nuclear options'.

Het hergebruik van door jou opgestelde requirements wordt van groter belang. Je zult langer met dezelfde materie bezig zijn, dus: breng efficiëntie in je werk. Met het opstellen van requirements houd je er meer rekening mee dat je ze later nodig hebt om weer in te kijken en her te gebruiken waar mogelijk.

T-shaped profiel

Met Agile waren we al begonnen om veelzijdiger te worden, je keek al wat vaker naar wat je buurman/-vrouw doet en hielp elkaar zo veel mogelijk. Met DevOps wordt dit alleen maar meer, er zitten meer verschillende expertises in het team, dus je kunt je met meer dingen bezig houden. En omdat een DevOps team continu zal blijven bestaan, ook in tijden waar er weinig wordt gedaan met requirements, zul je als requirements engineer ook meer tijd besteden aan andere taken. Denk bijvoorbeeld aan testen, functioneel beheer, maar misschien ook incident management omdat je de applicatie heel goed snapt.

Hier ga je de nodige cursussen voor volgen, en zo ontwikkel je een T-shaped profiel. Dat betekent dat je op enkele gebieden diepgaande kennis hebt (het verticale deel van de T, vooral requirements dus) maar ook enige kennis hebt op andere gebieden (het horizontale deel van de T). Let wel op dat je je eigen specialisme behoudt, en je daar ook in blijft ontwikkelen.

Leer je (OPS)buren kennen

Werkend in een DevOps-team gaat je je OPS-collega's steeds beter begrijpen. Hoewel je deze collega's in het verleden al betrokken in het opstellen van requirements, krijg je nu beter inzicht in hun werk omdat je letterlijk naast hen zit. Waar houden OPS'ers zich de hele dag mee bezig, en wat voor soort tools gebruiken ze?

Als requirements engineer merk je dan wat de noodzaak is van bijvoorbeeld goede monitoring. Daar maak je je weer hard voor bij de product owner, en daarna stel je samen met de teamleden hier goede requirements voor op. Ook denk ik dat je meer respect krijgt voor het brede, diepgaande en onvoorspelbare werk dat OPS'ers doen.

De klant nog beter begrijpen

In een DevOps team kun je niet om incidenten en andere verstoringen heen. Dit heeft zo z'n nadelen (bijvoorbeeld standby diensten draaien in het weekend), maar ook voordelen: je ziet echt waar dingen mis gaan. Waar loopt de klant tegenaan (bijvoorbeeld X keer per maand dat het systeem offline is), en welke gevolgen heeft dat? Dat geeft mooi inzicht in wat er verbeterd moet worden en wat hoge prioriteit heeft. Het laat soms ook pijnlijk zien waar je fouten in je requirements hebt gemaakt, maar daar leer je dan ook weer van.

Spelen met functionaliteit

In mijn zoektocht naar DevOps kom ik ook allerlei nieuwe termen tegen. Wat ik interessant vind is bijvoorbeeld 'dark launching' en 'levers & nuclear options'. Hiermee kun je nieuwe functionaliteit laten schaduwdraaien, of functionaliteit voor een bepaalde groep gebruikers actief maken om uit te proberen. Hoe gaaf is dat? Je test dan met echte gebruikers (die je hiervoor misschien nooit zou kunnen benaderen) en als het niet goed gaat deactiveer je de functionaliteit.

Je wordt er veelzijdiger van omdat je bij alle voorkomende taken in het team wordt betrokken en je leert meer van collega's.

Tenslotte weet je eigenlijk pas als de software gebruikt wordt of je requirements goed (genoeg) waren, en op deze manier kun je ze daarna nog makkelijk bijstellen.

Wat misschien wel even wennen is, is dat er soms meerdere oplossingen voor een behoefte worden gebouwd en dat de oplossing met meerdere kleine gebruikersgroepen wordt geprobeerd en getest. Maar eigenlijk is dat wel heel gaaf, want jij wilt toch ook dat de beste oplossing in productie wordt genomen?

Dus wat betekent het nou?

Het werken in een DevOps team is voor een requirements engineer niet slechts 'vanaf nu ga ik beheer meer betrekken', nee het behelst veel meer voor de werkzaamheden in deze rol.

Je bent namelijk nog meer deel van de organisatie geworden omdat je in een continu werkend lijnteam werkt. Je wordt er veelzijdiger van omdat je bij alle voorkomende taken in het team wordt betrokken en je leert meer van collega's. Wat ook erg nuttig en zinvol is, is dat je de klant veel beter gaat begrijpen door de extra informatie die je krijgt en dat er allerlei technieken ingezet kunnen worden om meer met functionaliteit te spelen en je zo de beste oplossing voor de klant kunt kiezen. Eigenlijk is DevOps dus een prachtige verrijking van de functie requirements engineer!

