
Agile werken, Service Level Manager staat buiten spel

Met de komst van Agile werken worden bij menig organisatie behoorlijke organisatieveranderingen doorgevoerd. De vraag is in hoeverre de controls van beheer daar nog wel de juiste aandacht krijgen. Dit artikel kijkt daartoe vooral naar het service level management proces in combinatie met Scrum.

Inleiding

Scrum is een Agile ontwikkelproces dat steeds meer invulling krijgt. De product owner vertegenwoordigt de gebruikersorganisatie. Hij is de eigenaar van de product backlog waarop de gewenste veranderingen zijn opgenomen en bepaalt hiervan ook de prioriteit. Het development team realiseert in sprints de gewenste toepassing en wordt gecoacht door de Scrum master. De product owner accepteert namens de stakeholders de gerealiseerde producten. En daarbij eindigt de scope van het Scrum team.

Een belangrijke vraag is wat nu de relatie is met de service level manager. Is die rol nog wel nodig? En zo ja hoe kan die dan worden ingevuld? En wat is de huidige best practice?

Noodzaak service level manager

De product owner bepaalt zowel de functionaliteit als de kwaliteit van de gewenste applicatie. Vanuit die hoedanigheid vervult hij eigenlijk een soort van service level management rol tussen de stakeholders en het Scrum team. Scrum spreekt echter niet over een SLA en stelt dat interactie met de gebruiker belangrijker is dan contracten en dus ook SLA's. Als de product owner gedurende de gehele levenscyclus van de applicatie het Scrum team aanstuurt en daarbij geen andere partijen zijn betrokken zoals een applicatiebeheer team of infrastructuur team dan zou dat model kunnen werken. Maar in de praktijk zijn applicaties veelal opgenomen in een keten, moet de applicatie beheerd worden en zijn er infrastructurele voorzieningen nodig om de applicatie operationeel te krijgen en houden. Ook zijn veel product owners niet verantwoordelijk voor de gehele levenscyclus van de applicatie. Zelfs met het introduceren van een DevOps team dat niet alleen de ontwikkeling doet maar ook het beheer van de applicatie geldt dat er vaak vele afhankelijkheden zijn die afspraken vereisen met andere partijen.

Invulling rol service level manager

De product owner lijkt geknipt te zijn voor deze rol. Maar hij is veelal de vertegenwoordiger van de gebruikersorganisatie. Daarbij kan hij niet tevens de rol vervullen van de ICT-organisatie. Dan zou hij zijn eigen leverancier worden.

Een betere optie is de oude rol van service level manager te hanteren. De service level manager spreekt dan een SLA af met de product owner waarbij de product owner de non-functionele requirements bepaalt en de service level manager binnen de ICT-organisatie afspraken maakt om deze requirements te borgen. Deze borging wordt dan met zowel externe (underpinningcontracts (UC)) als interne leveranciers (Operational Level Agreement (OLA)) afgesproken. Het is belangrijk om te onderkennen dat het development team zelf ook een leverancier is. Met dit team moeten ook afspraken gemaakt worden, bijvoorbeeld door non-functional requirements op de product backlog te plaatsen en kwaliteitsmaatregelen op de Definition of Done te plaatsen.

Huidige best practice?

Als ik vanuit mijn eigen ervaring spreek, dan is de rol van service level manager zeker niet uitgespeeld. Wat echter veelal gebeurt, is dat de service level manager niet betrokken is bij de totstandkoming van een nieuwe applicatie. Het Scrum team gedraagt zich dus als een externe leverancier die rechtstreeks communiceert met de klant zonder afstemming met de rest van de serviceorganisatie. En dat is nu net wat we niet zouden moeten willen. Daarmee staat de service level manager namelijk buitenspel en moet roeien met de riemen die hem aangereikt worden. Dit resulteert in gescheiden werelden van ontwikkeling en beheer waardoor de uiteindelijke kwaliteit van de serviceverlening suboptimaal is.

Bart de Best

Auteur van het boek 'Agile Service Management met Scrum' en trainer bij [IT Management Group](#) voor onder andere de training [Masterclass Agile Service Management met Scrum](#).