"Een must read voor iedere analist die zich verder wil professionaliseren."

> **Marcel Schaar, voorzitter International Institute of Business Analysis (IIBA) Nederland**

"Aanrader voor analisten die de (nieuwe) requirementstechnieken met succes willen toepassen."

> **Bram Walet, voorzitter Requirements Kenniscentrum**

"Met dit boek hebben alle leden van een agile team toegang tot de kennis die nodig is om samen het juiste product te ontwikkelen."

> **Patrick Verheij, voorzitter Agile Consortium Nederland**

"Handboek Requirements biedt een compleet, actueel en handzaam overzicht van het vakgebied, zowel voor studenten als professionals."

> **Wiebe de Witte, consultant en docent Business IT & Management, Hogeschool van Amsterdam**

"Dit is niet alleen zeer prettig lesmateriaal maar voor studenten ook ideaal als naslagwerk bij het afstuderen."

> **Gerda In 't Veld, docent HBO-ICT Software Engineering, Haagse Hogeschool**

"Een zeer toegankelijk en compleet boek zowel voor studenten als lectoren/docenten. Aanrader voor iedereen die het requirementsvak wilt leren!"

> **Margot den Donder, opleidingsvoorzitter Toegepaste Informatica, Hogeschool Gent**

"Sommige requirements weet je vooraf, andere ontdek je nog gaandeweg. Dus ook als je requirements niet gedetailleerd van te voren opschrijft, ze zijn er wel. Dit boek helpt je de weg te vinden in dit ingewikkelde maar interessante vakgebied."

> **Rini van Solingen, auteur van de bestseller *De Kracht van Scrum***

Handboek Requirements

Leidraad voor analisten in agile, traditionele en hybride omgevingen

Nicole de Swart

eburon
business

ISBN 978 94 6301 111 2

Eburon Business
Postbus 2867
2601 CW Delft
tel.: 015-2131484
info@eburon.nl / www.eburonbusiness.nl

Omslagontwerp: Korona
Opmaak binnenwerk: Nicole de Swart

Eerste druk, juli 2010
Geheel herziene druk, augustus 2017

Inhoud

Inleiding

Requirements vervullen binnen de softwareontwikkeling een belangrijke rol; daar zijn de meeste ICT'ers het wel over eens. Ook binnen agile ontwikkeling nemen requirements een prominente plaats in. Toch is het werk er voor requirementsanalisten de laatste jaren niet makkelijker op geworden. Er is veel veranderd in het vakgebied en in de rol en verwachtingen van de requirementsanalist. Er zijn bovendien ontelbaar veel nieuwe requirementstechnieken en best practices bijgekomen.

Hoog tijd dus voor een grondige update en actualiseringsslag van Handboek Requirements. Bij het trainen en helpen van vakgenoten bij het toepassen van requirementstechnieken, heb ik gezien hoe belangrijk het is om als requirementsanalist zowel de agile als de traditionele requirementsmethoden en -technieken onder de knie te hebben. Dan ben je in staat om je werkwijze af te stemmen op de veranderende omgeving waarin je werkt.

Dit handboek is geschreven voor diegenen die het complete vakgebied willen doorgronden en vooral voor analisten die in de dagelijkse praktijk met requirements werken. Het biedt een leidraad voor analisten in agile, traditionele en hybride omgevingen. De opbouw van het boek maakt het mogelijk om snel informatie en praktische tips over specifieke onderwerpen te vinden. Bij integrale lezing neemt Handboek Requirements je mee door het vakgebied en de toepassing ervan.

Naast het actualiseren en verbeteren van de bestaande tekst, is aan deze geheel herziene uitgave vooral veel informatie over agile requirements toegevoegd. Om de compleet nieuwe hoofdstukken en uitbreidingen op bestaande hoofdstukken te integreren, is de opbouw en indeling van het boek aangepast. Het boek bestaat nu uit vier delen, voorafgegaan door een hoofdstuk dat een overzicht van het totale vakgebied geeft. Deel I gaat uitgebreid in op de diverse typen requirements. Deel II gaat in op de aanpak die de requirementsanalist volgt en maakt daarbij onderscheid tussen agile en meer traditionele trajecten. Deel III gaat dieper in op het uitwerken van requirements en legt de twee meest gebruikte requirementstechnieken uit. Tot slot behandelt deel IV een breed scala aan agile en traditionele requirementstechnieken.

Deze nieuwe uitgave was er niet gekomen als er niet zoveel requirementsanalisten dit boek zouden lezen en als naslagwerk gebruiken. Als blijk van dank geef ik mijn lezers graag iets extra's. Op www.handboekrequirements.nl/lezers-e-book staat het e-book Agile Requirements klaar. Dit kun je, als lezer, downloaden met behulp van code REQ7.

Omwille van de leesbaarheid spreekt dit boek over 'hij'. Hiervoor mag vanzelfsprekend ook 'zij' gelezen worden.

Hoofdstuk 1
Vakgebied in beweging

Het requirementsvakgebied heeft de afgelopen decennia een aanzienlijke ontwikkeling doorgemaakt. Dit inleidende hoofdstuk start met de kern van het vakgebied requirements engineering. Daarna volgen de verschuivingen die daarin hebben plaatsgevonden en de invloed die agile op het requirementsvak heeft. De weerslag die dat op de rol en verantwoordelijkheden van de requirementsanalist heeft gehad, staat in het laatste deel van dit hoofdstuk.

Requirements Engineering

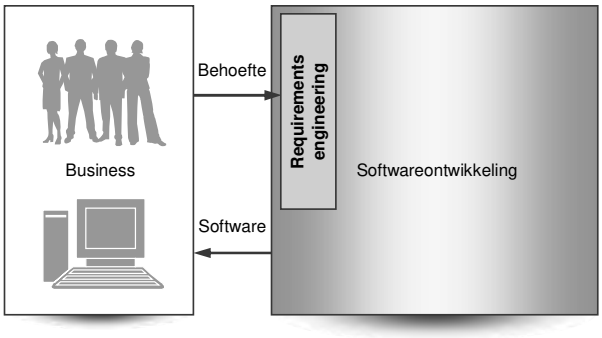
Requirements engineering is een discipline binnen de softwareontwikkeling. Voordat een organisatie besluit om een softwareontwikkeltraject te starten is doorgaans uit een bedrijfs- of een marktonderzoek gebleken dat er behoefte is aan extra geautomatiseerde ondersteuning. Dit kan bijvoorbeeld zijn omdat er aanpassingen in de interne bedrijfsprocessen nodig zijn, omdat een bestaand systeem verouderd is of omdat er mogelijkheden zijn voor een nieuw commercieel softwareproduct. Requirements engineering gaat over het concretiseren van de behoefte van de business en de gewenste geautomatiseerde ondersteuning daarbij. Dit is weergegeven in Figuur 1. De belanghebbenden uit de business willen een systeem[1] dat voldoet aan hun behoeften. De andere disciplines binnen softwareontwikkeling nemen die behoeften, de requirements, als basis.

[1] Een systeem is in dit boek een geautomatiseerd systeem vanuit het gezichtspunt van de business. In IT-georiënteerde organisaties denkt men bij 'systeem' eerder aan het IT-systeemlandschap en de daarbinnen vanuit IT-oogpunt gedefinieerde systemen.

> Het doel van requirements engineering is het tot stand brengen en in stand houden van overeenstemming tussen de opdrachtgever, de overige belanghebbenden uit de business en het softwareontwikkelteam over de requirements.

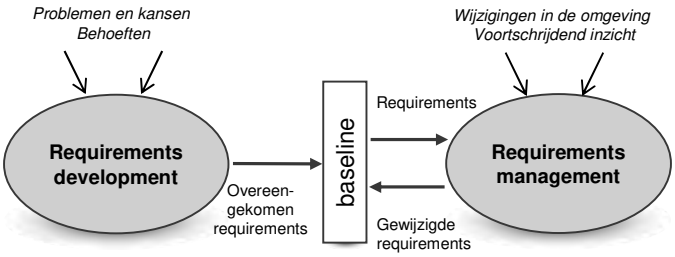
Overeenstemming tussen de opdrachtgever, de overige belanghebbenden uit de business en het softwareontwikkelteam over de requirements is uitermate belangrijk. Dit geldt niet alleen op hoofdlijnen maar ook op detailniveau. De opdrachtgever is eindverantwoordelijk, maar laat de afstemming over de gedetailleerde requirements gewoonlijk over aan andere belanghebbenden. Zij vertegenwoordigen de opdrachtgever. De belangen van de overige belanghebbenden kunnen onderling verschillen en conflicteren met de doelen van de opdrachtgever. Het gaat bij requirements engineering daarom om het bereiken van overeenstemming tussen de opdrachtgever, de overige belanghebbenden uit de business en het softwareontwikkelteam over de requirements op alle detailniveaus.

Een voorwaarde voor het bereiken van overeenstemming is vanzelfsprekend dat alle belanghebbenden de requirements goed en op dezelfde manier begrijpen. Anders lijkt er overeenstemming te zijn, maar is die er in werkelijkheid niet.



Figuur 1: Requirements engineering als onderdeel van softwareontwikkeling

Requirements engineering omvat zowel het tot stand brengen als het in stand houden van overeenstemming. Het tot stand brengen van overeenstemming over de requirements staat bekend als requirements development. Dit resulteert in een verzameling goedgekeurde requirements, baseline genoemd. De baseline vormt de basis voor de softwarewareontwikkeling. Daarna is aandacht nodig voor het in stand houden van de overeenstemming. De requirements in de baseline zijn namelijk niet in beton gegoten. Wanneer er tijdens het project wijzigingen in de eerder overeengekomen en misschien al geïmplementeerde requirements wenselijk zijn, moet het mogelijk zijn om de requirements aan te passen. Het onderdeel van requirements engineering dat zich richt op het gecontroleerd doorvoeren van wijzigingen in de requirements heet requirements management. In Figuur 2 is dit schematisch weergegeven.



Figuur 2: Requirements engineering

Het vakgebied softwareontwikkeling en als gevolg daarvan ook requirements engineering hebben de afgelopen decennia een aanzienlijke ontwikkeling doorgemaakt. De aandacht verschoof van het systeem- naar het businessperspectief.

Systeemperspectief

In de jaren zeventig van de vorige eeuw kwam men tot het inzicht dat de gewenste acties van het systeem (de functionaliteit) en de implementatie daarvan (de technische oplossing) afzonderlijk bekeken moeten worden. Door 'het wat' van 'het hoe' te scheiden, kwam er expliciet aandacht voor requirements en ontstond het nieuwe vakgebied requirements engineering. In de eerste twee decennia stonden het systeem en de eisen die de business daaraan stelt centraal. Deze eisen werden softwarerequirements genoemd en zijn onder te verdelen in functionele en niet-functionele softwarerequirements. Hoofdstuk 2 Requirementsperspectieven en vooral Hoofdstuk 5 Softwarerequirements gaan hier uitgebreid op in.

Het was gebruikelijk om requirements als atomaire beweringen te beschrijven. Iedere bewering bevat dan precies één eenduidig geformuleerde requirement. Voor een gemiddeld systeem ging het al snel om honderden gedetailleerde softwarerequirements die in Excel of in een requirementsmanagementtool beheerd werden. Daaruit genereerde de requirementsanalist het product softwarerequirementsspecificatie (SRS). Hoofdstuk 8 Requirementsproducten gaat in op het SRS. Ondanks een requirementsmanagementtool was het lastig om overzicht te houden en de vele individuele requirements in context te plaatsen. Daardoor traden gemakkelijk interpretatieverschillen op en waren de requirements moeilijk te valideren. Naast dit systeemperspectief op requirements ontstond in de jaren negentig het businessperspectief.

Businessperspectief

Met de komst van het businessperspectief verschoof het accent van de functionaliteit van het systeem naar de bedrijfs- en klantprocessen die het ondersteunt. Het systeem ontleent haar bestaansrecht immers aan de toegevoegde waarde die het levert aan de business. Het is daarom beter om de requirements vanuit de te ondersteunen processen te benaderen. Hierdoor blijft de aandacht van het softwareontwikkelteam gericht op de toegevoegde waarde die het systeem moet leveren. Een systeem is immers geen

doel op zich. Bij het businessperspectief staan de behoeften van de business aan geautomatiseerde ondersteuning centraal. Dit perspectief bevat twee typen requirements: business- en gebruikersrequirements. Hoofdstuk 3 Businessrequirements en Hoofdstuk 4 Gebruikersrequirements zijn volledig aan deze requirementstypen gewijd.

Use cases zijn in de negentiger jaren uitgegroeid tot een populaire techniek voor het specificeren van de requirements. Een use case geeft aan hoe het systeem en de gebruiker samenwerken om een eindresultaat te halen dat waarde heeft voor de gebruiker. Hoofdstuk 4 Gebruikersrequirements en Hoofdstuk 12 Use case 2.0 lichten deze techniek toe. Vanaf de jaren negentig gingen veel softwareontwikkelprojecten use case-gedreven werken. Dit betekent dat alle disciplines binnen het project de use cases als werk- en communicatie-eenheden gebruiken. Dit geldt voor de planning en de voortgangsbewaking, voor het achterhalen en vastleggen van requirements, voor het ontwerpen, het ontwikkelen en testen van de software. Een groot voordeel van use case-gedreven werken is dat alle betrokkenen, van opdrachtgever en gebruikers tot ontwikkelaars en testers, een gezamenlijke 'kapstok' hebben. Iedereen praat over dezelfde dingen, de use cases. Voor die tijd maakte iedere discipline een vertaalslag naar zijn eigen werkeenheid en was de samenhang vaak niet inzichtelijk (Baker, 2003).

Agile gedachtengoed

Rond de eeuwwisseling is agile softwareontwikkeling ontstaan. Agile hanteert een andere visie op softwareontwikkeling en is een reactie op de document- en plangedreven, zware softwareontwikkelmethoden die tot dan toe gebruikt werden. In 2001 hebben de voorlopers op dit gebied gezamenlijk een manifest opgesteld met daarin de belangrijkste principes voor agile softwareontwikkeling (Agile Alliance, 2001). Letterlijk vertaald betekent *agile* wendbaar, beweeglijk of vlug. Agile softwareontwikkeling maakt het mogelijk in te spelen op veranderingen en om te gaan met onzekerheden.

> De agile aanpak brengt softwareontwikkeling terug tot de kern, namelijk het ontwikkelen van voor de business waardevolle software.

Het hele agile team is gericht op het leveren van zo veel mogelijk waarde voor de business. Een agile team bestaat gewoonlijk uit een product owner, een multidisciplinair ontwikkelteam en een scrum master (zie volgende paragraaf). Ze werken in iteraties (*sprints*) van maximaal vier weken, waarin ze steeds nieuwe functionaliteit aan het systeem toevoegen. Aan het einde van iedere iteratie levert het ontwikkelteam, in technische en bij voorkeur ook in functionele zin, productierijpe software op. Ze beginnen met een heel eenvoudige variant van het systeem en verbeteren en breiden die iteratie na iteratie uit. Op deze manier toetsen ze of de software voldoet aan de verwachtingen van de belanghebbenden en krijgt de opdrachtgever de mogelijkheid om de software vroegtijdig in gebruik te nemen. Hoofdstuk 6 Agile softwareontwikkeling gaat hier verder op in.

Het uitgebreid en nauwkeurig specificeren van requirements aan het begin van het softwareontwikkeltraject is in agile omgevingen overbodig en onwenselijk. Een pro-

ductvisie met het bedrijfsdoel en een opsomming van de voornaamste kenmerken van het systeem geeft voldoende richting. Gebruikers kunnen namelijk vooraf niet exact aangeven wat ze nodig hebben. Bovendien zijn wijzigingen in de requirements onvermijdelijk vanwege voortschrijdend inzicht en veranderende omstandigheden.

> Pas wanneer gebruikers het systeem zien of ermee werken, komen ze erachter welke geautomatiseerde ondersteuning ze precies nodig hebben.

Hierdoor neemt terugkoppeling op de zojuist ontwikkelde software een belangrijke plaats in binnen agile. De terugkoppeling en de ervaringen met de tot dan toe opgeleverde software, zijn essentiële input voor de requirements op de product backlog. Een product backlog is een geprioriteerde opsomming van nog uit te werken en te implementeren requirements. Het is een opsomming die voortdurend geactualiseerd wordt om in lijn te blijven met de wijzigende inzichten. Meer informatie over de product backlog is opgenomen in Hoofdstuk 8 Requirementsproducten.

Het uitwerken van de in de product backlog opgesomde requirements vindt plaats kort voor de iteratie waarin het ontwikkelteam ze nodig heeft. Het hele agile team en eventueel extra vertegenwoordigers van de business nemen actief deel aan speciaal daarvoor bedoelde bijeenkomsten, die elke iteratie terugkeren. Het is dan voor het ontwikkelen van de software niet nodig om requirements uitgebreid, volledig en eenduidig vast te leggen. Het opstellen van beheerdocumentatie doet het team namelijk tijdens de iteraties. In Hoofdstuk 6 Agile softwareontwikkeling zijn deze activiteiten nader beschreven.

User stories zijn binnen agile veruit de meest gebruikte requirementstechniek. Ze maken het de belanghebbenden uit de business en het ontwikkelteam mogelijk om de requirements op het juiste moment voor iedereen inzichtelijk te maken. User stories hebben een vaste zinsopbouw die helpt om de requirements vanuit het gezichtspunt van de gebruiker te bekijken en de aandacht te richten op de toegevoegde waarde voor de gebruiker. Meer hierover in Hoofdstuk 4 Gebruikersrequirements en Hoofdstuk 11 User stories.

Requirements engineering en agile

Zoals beschreven in de eerste paragraaf draait het bij requirements engineering om het vaststellen van een baseline met overeengekomen requirements. De requirementsanalist is verantwoordelijk voor het vullen van de baseline met eenduidig gespecificeerde en goedgekeurde requirements. Deze specificaties mogen daarna niet zomaar gewijzigd worden. De requirementsanalist is verantwoordelijk voor het beheer ervan.

Hoewel een deel van de requirements engineeringstechnieken ook zeker in een agile omgeving toepasbaar zijn, is de kern en gedachtegang van het vakgebied requirements engineering fundamenteel anders dan dat van agile. Sterker nog, requirements worden binnen agile niet als afzonderlijk vakgebied onderscheiden. Requirements en bijvoorbeeld ook testen maken integraal onderdeel uit van het agile proces (zie hiervoor Hoofdstuk 6 Agile softwareontwikkeling). Of je bij agile nog kunt spreken over

het vakgebied requirements engineering is voor discussie vatbaar.

> Vanwege de grote verschillen in gedachtegang, producten en werkwijze, spreekt dit boek binnen agile niet over requirements engineering. Het vakgebied requirements engineering behoudt haar oorspronkelijke betekenis en duidt op een traditionele, niet-agile omgeving.

Requirements engineering is opgebouwd uit requirements development en requirements management. Bij requirements development draait het om het achterhalen, analyseren, specificeren en valideren van de requirements. Deze staan bekend als de kennisgebieden van requirements development en komen in Hoofdstuk 15 Requirements developmenttechnieken uitgebreid aan bod. De requirementsanalist maakt zo het gewenste eindresultaat, het te ontwikkelen systeem, inzichtelijk. Daarna volgt requirements management. Wijzigingen moeten namelijk gecontroleerd gebeuren om te voorkomen dat de omvang van het systeem ongemerkt toeneemt en dat er onduidelijkheid ontstaat over de actuele inhoud van de overeengekomen requirements. Aangezien iedere wijziging tijd en geld kost, kunnen niet alle wijzigingsverzoeken zonder meer doorgevoerd worden. Een zorgvuldige kosten-baten afweging is noodzakelijk.

Bij agile zit het achterhalen van requirements veel meer verweven in het ontwikkelproces. Gedurende het ontwikkeltraject helpen de product owner en het ontwikkelteam de belanghebbenden uit de business te ontdekken welke systeemondersteuning ze precies nodig hebben. Daarbij is de reeds ontwikkelde software en de terugkoppeling vanuit de business daarop, een onmisbaar instrument. Het streven is niet om meteen de juiste requirements te vinden, maar om gaandeweg het traject steeds beter zicht te krijgen op de werkelijke behoefte van de business.

> Een agile team gaat ervan uit dat zij (en ook de business) nog niet weten aan welke requirements het uiteindelijke systeem moet voldoen. Bij requirements engineering probeert de requirementsanalist in één keer de juiste requirements in kaart te brengen.

De baseline in requirements engineering en de product backlog in agile zijn niet met elkaar te vergelijken. De baseline moet het gewenste eindresultaat weerspiegelen. Het moet precies aangeven hoe het te ontwikkelen systeem eruit komt te zien, zodat de benodigde werkzaamheden op basis daarvan ingepland en uitgevoerd kunnen worden. Daarentegen is een product backlog meer een actielijst. Een lijst van nog uit te werken en met behulp van werkende software te valideren requirements.

> De items op een product backlog zijn te beschouwen als geheugensteun voor nog uit te voeren (requirements)werk. De items in de baseline zijn goedgekeurde specificaties van requirements die aan kwaliteitscriteria zoals eenduidigheid en volledigheid moeten voldoen.

Dit boek beschrijft hoe agile teams met requirements omgaan en, meestal in afzonderlijke hoofdstukken, hoe traditionele requirementsanalisten met requirements omgaan. In de praktijk komen veel mengvormen van de agile en traditionele aanpak voor. Een organisatie of project heeft dan bewust of soms onbewust elementen uit beide aanpakken gecombineerd tot een eigen aanpak op maat. Bij verschillende onderwerpen besteedt dit boek expliciet aandacht aan deze hybride aanpakken.

Requirementsanalist

Bij een vakgebied in beweging is het niet verwonderlijk dat ook de voornaamste rol daarbinnen mee verandert. Bovendien is er binnen het vakgebied geen overeenstemming over de rolnaam van degene die de requirements opstelt. De meest gebruikte rolnamen binnen requirements engineering zijn requirementsanalist, requirementsengineer, requirementsmanager, informatieanalist en businessanalist. Deze rolnamen komen door elkaar en in verschillende betekenissen voor. Ze zijn niet beschermd en iedereen mag zelf de inhoud ervan definiëren. De naam businessanalist komt uit Amerika. Daar is *business analyst* de meest gebruikte benaming. Requirements engineering is daar een onderdeel van. In Nederland was lange tijd de rolnaam informatieanalist bijzonder populair. Het bekendste certificeringsprogramma in Nederland, van het International Requirements Engineering Board (IREB), hanteert requirementsengineer als rolnaam.

Dit boek hanteert de rolnaam requirementsanalist voor degene die het opstellen van requirements als taak heeft. In een softwareontwikkeltraject kunnen dit een of meerdere personen zijn. Voor het gemak hanteert dit boek overal requirementsanalist in enkelvoud.

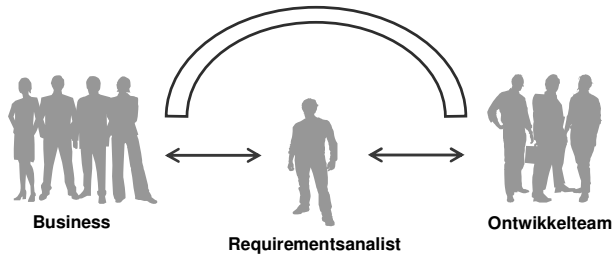
De onderstaande paragrafen geven in grote lijnen de taken en verantwoordelijkheden van een requirementsanalist aan, binnen requirements engineering en binnen agile softwareontwikkeling.

Requirements engineering

Een of meer personen nemen impliciet of expliciet de rol van requirementsanalist op zich. De requirements moeten immers terecht komen bij het softwareontwikkelteam. Dit gebeurt soms doordat de business zelf haar requirements noteert en aan een interne of externe opdrachtnemer overhandigt. Het komt ook voor dat iemand uit het softwareontwikkelteam, bijvoorbeeld een ontwikkelaar of een softwarearchitect, gaat uitzoeken aan welke requirements het systeem moet voldoen. Het is echter aan te bevelen om iemand die zich heeft gespecialiseerd in requirements engineering expliciet de rol van requirementsanalist te geven. Het specificeren van requirements is immers niet eenvoudig en is een kritieke succesfactor voor softwareontwikkeltrajecten gebleken (Forrester, 2006 en Standish Group, 2009).

De requirementsanalist vervult een brugfunctie tussen de business en het softwareontwikkelteam. Hij helpt de belanghebbenden uit de business bij het definiëren van hun

requirements en brengt deze over aan het ontwikkelteam. Dit zijn twee totaal verschillende groepen met elk hun eigen achtergrond. De requirementsanalist brengt deze twee werelden bij elkaar. Hij weet welke informatie het ontwikkelteam nodig heeft om een systeem te realiseren. Hij zoekt samen met de belanghebbenden uit de business naar de behoeften die zij hebben aan geautomatiseerde ondersteuning. In Figuur 3 is de brugfunctie van de requirementsanalist gevisualiseerd.



Figuur 3: Brugfunctie requirementsanalist

Het is de verantwoordelijkheid van de requirementsanalist om overeenstemming over de requirements tussen de belanghebbenden uit de business en het softwareontwikkelteam, tot stand te brengen en in stand te houden. De requirementsanalist legt de overeengekomen requirements vast in requirementsproducten (zie Hoofdstuk 8 Requirementsproducten). Het softwareontwikkelteam implementeert vervolgens de vastgelegde requirements. Het opstellen van de requirementsproducten wordt vaak gezien als dé verantwoordelijkheid van de requirementsanalist. Het gaat echter niet om de vastlegging, maar om het achterhalen en overbrengen van de requirements. De requirementsproducten zijn niet meer dan een communicatiemiddel.

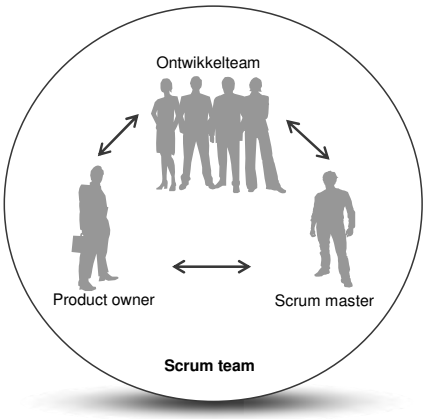
De requirementsanalist houdt zich bezig met alle requirements inclusief de niet-functionele requirements. Dit type requirements is vaak onderbelicht en wordt soms gezien als een technische aangelegenheid die door de softwarearchitect uitgezocht moet worden. Hoofdstuk 13 Niet-functionele requirements laat zien hoe de requirementsanalist deze requirements specificeert.

Agile softwareontwikkeling

Agile werkt met multidisciplinaire ontwikkelteams van ongeveer zeven personen. Die personen werken intensief samen en zijn als team verantwoordelijk voor het opleveren van het systeem. De teamleden bezitten gezamenlijk alle competenties die vereist zijn om het gewenste resultaat te realiseren. Er bestaan geen afzonderlijke rollen. Van teamleden wordt verwacht dat ze ook (meehelpen bij) werkzaamheden die buiten hun specialisme vallen, uitvoeren. Om dat te benadrukken heeft niemand in een agile team de rol van bijvoorbeeld tester of requirementsanalist. Iedereen is gewoon teamlid.

> Een agile team bestaat gewoonlijk uit een multidisciplinair ontwikkelteam, een product owner en een scrum master.

In Figuur 4 zijn de drie scrumrollen weergegeven. Samen vormen ze het complete agile team, scrum team genoemd. De *scrum master* zorgt er als dienend leider voor dat iedereen binnen en buiten het scrum team de agile principes en regels begrijpt en daarnaar handelt. De andere rol, die van *product owner*, neemt in het kader van requirements een vooraanstaande positie in. De product owner is verantwoordelijk voor het maximaliseren van de businesswaarde van het te ontwikkelen systeem en van het werk van het ontwikkelteam. Daarmee is hij verantwoordelijk voor het managen van de product backlog (Schwaber & Sutherland, 2016). De product owner kan bij het opstellen van de requirements in de product backlog zelf de leiding nemen of het werk aan het ontwikkelteam uitbesteden. In beide gevallen is samenwerking noodzakelijk en blijft de product owner verantwoordelijk. In het ontwikkelteam moeten de competenties om requirements te achterhalen aanwezig zijn. In de praktijk maakt daarom vaak een requirementsanalist deel uit van het ontwikkelteam.



Figuur 4: De rollen binnen een scrum team

De overlap tussen de rollen van de product owner en van de requirementsanalist binnen requirements engineering is beperkt. De product owner is qua verantwoordelijkheden eerder een combinatie van de productmanager, projectmanager en requirementsanalist. Het is een belanghebbende uit de business, geen IT'er. Zijn doel is om bijvoorbeeld een bedrijfsprobleem op te lossen of een commerciële kans te benutten en weet (of denkt te weten) wat voor systeem daarvoor nodig is. De product owner draagt zijn visie uit, stelt prioriteiten, neemt beslissingen, managet de verwachtingen en maakt helder wat het te ontwikkelen systeem moet kunnen. Dit kan hij alleen met succes doen als hij samenwerkt met en gebruikmaakt van de kennis en ideeën van het ontwikkelteam en van de gebruikers(vertegenwoordigers).

> Het blijkt vaak lastig om een belanghebbende uit de business te vinden die de rol van product owner kan en wil vervullen. In dat geval valt de organisatie meestal terug op een requirementsanalist die dan optreedt als of namens de product owner.