

# Als requirements engineer tussen twee werelden in

Dit DREAMagazine heeft als thema *Tussen twee werelden*, waar we het vooral hebben over de wereld van Agile & de wereld van Waterval. Maar als requirements engineer zit je eigenlijk altijd tussen twee andere werelden in, ongeacht of je volgens een waterval of een iteratieve methode werkt. We bevinden ons tussen de wereld van de gebruiker en de wereld van de ontwikkelaar, en dat zijn twee heel verschillende werelden.

door Linda Haak - van der Spek

## De gebruiker:

De gebruiker is degene die het grote belang heeft bij het verandertraject. Je kunt hem in dit artikel lezen als de opdrachtgever of senior user. Hij wil een nieuw product, systeem en/of aanpassingen daarop, om met zijn organisatie mee te gaan in marktontwikkelingen/wetswijzigingen of veranderende omstandigheden. De gebruiker werkt meestal in meer projecten tegelijk en voert vaak ook nog lijnactiviteiten uit. Het "gewone" werk moet tenslotte ook doorgaan.

Communiceren, inlevingsvermogen en belangen behartigen zijn de kerncompetenties in dit werk. Als je die beheerst vlieg je in een vloeiende lijn heen en weer tussen de diverse werelden.

## De ontwikkelaar:

De ontwikkelaar, ook wel bekend als programmeur, bouwt de gewenste (aanpassingen in de) applicatie. Zijn voornaamste input komt van de requirements engineer en architect. De ontwikkelaar werkt soms aan één project, soms aan meer projecten tegelijk.

De gebruiker en de ontwikkelaar spreken verschillende talen. Ook hebben ze een andere visie op hoe de wereld eruit moet zien en hebben ze verschillende prioriteiten. Dat levert nog weleens frictie op. Als requirements engineer ben je de vertaler en bemiddel je tussen de verschillende werelden. Waar zitten die verschillen in en waar moet je als requirements engineer rekening mee houden? Hier een aantal voorbeelden:

- 1 De (buiten)wereld versus het IT-landschap.** Een gebruiker heeft (als het goed is) een beeld van de wereld binnen en buiten het bedrijf, en waar het product en/of het bedrijf naar toe moet. Dit beeld is de belangrijkste input voor de gesprekken die je als requirements engineer voert. De ontwikkelaar heeft een heel andere wereld, namelijk die van de applicaties waaraan hij werkt. Het IT landschap is zijn wereld. Deze wereld brengt een andere visie en andere prioriteiten met zich mee dan de wereld van de gebruiker. Requirements engineers vertalen deze werelden. Dat betekent dat je de ontwikkelaar uit legt wat het business doel is van de verandering, zonder hem te overvoeren met onnodige details. Aan de andere kant moet je soms aan de gebruiker uitleggen waarom sommige dingen niet zomaar 'even' aangepast kunnen worden, omdat dit in de IT wereld niet een kwestie van 'even' is.
- 2 De onbekendheid van morgen tegenover de behoefte om alles mogelijk te maken.** De wereld van de gebruiker verandert. Dat betekent vaak dat hij vandaag nog niet kan bedenken, waar hij morgen op wil inspelen. De ontwikkelaar daarentegen wil graag al rekening houden met functionaliteiten die later toegevoegd moeten worden. Maar als deze niet bekend zijn, kan dit soms frustrerend zijn omdat hij zijn werk over een tijdje weg kan gooien, vanwege gewijzigd inzicht. Als requirements engineer kun je rekening houden met de (on)verwachte wijzigingen: bv door je ontwerpdocumentatie modulair op te bouwen, zodat deze gemakkelijk aan te passen en herbruikbaar is. Het is belangrijk om goed te communiceren waarom sommige dingen voor de ontwikkelaar niet in een logische volgorde worden verwezenlijkt. Ook is het belangrijk aan te geven waarom sommige inzichten zijn veranderd en waarom sommige dingen dus moeten worden weggegooid. Je moet de visie van de gebruiker duidelijk vertalen zodat dit logisch(er) in de oren klinkt bij de ontwikkelaar.

- 3 **Vershil in prioriteiten.** De gebruiker kijkt naar de business value, welke waarde gaat dit opleveren en wat is dan het belangrijkste en moet als eerste gedaan worden. De ontwikkelaar kijkt naar wat technisch juist makkelijk of moeilijk is. Daarom moet je als requirements engineer optreden als onderhandelaar tussen de gebruiker en de ontwikkelaar: wat is nu wijsheid om als eerste te ontwikkelen?
- 4 **Een simpel verhaal met 1 powerpoint versus een uitgewerkt detailontwerp.** Een gebruiker wil vooraf zo min mogelijk documenteren "dat moet ik dan allemaal weer opschrijven, je snapt toch wat ik bedoel, kunnen jullie niet gewoon ontwikkelen", terwijl ontwikkelaars juist vaak gebaat zijn bij meer details. Ontwikkelaars willen graag lezen wat er wel en niet moet gebeuren (tot een zekere hoogte). Als requirements engineer zoek je hier weer naar de middenweg; wat is goed genoeg voor beide partijen? Bij te veel details is niemand gebaat, maar als het niet voldoende duidelijk is wat de gebruiker wil, is de kans op een succesvol project klein.
- 5 **Een archipel aan stakeholders versus een eiland met een ontwerper.** De opdrachtgever heeft te maken met een grote diversiteit aan personen (stakeholders). Daar moet hij en de requirements engineer rekening mee houden. Voor de ontwikkelaar zijn er minder mensen in zijn wereld, zijn landschap is kleiner. De wereld van de ontwikkelaar lijkt eigenlijk meer een eiland of meer een eilandenkolonie als hij werkt aan verschillende projecten. Daarnaast lijkt het vaak alsof er niet veel gesproken wordt tussen die eilanden, wat de ontwikkelaar in zijn werk kan tegenwerken, omdat hij tegenstrijdige of niet op elkaar afgestemde opdrachten krijgt. De requirements engineer is hier wederom de spin in het web die de verschillende belangen moet behartigen, wat ook kan betekenen dat hij met requirements engineers uit andere projecten moet afstemmen.
- 6 **Gebruiker denkt vaak in beperkingen ("dat zal wel niet kunnen") terwijl sommige ontwikkelaars echte Bob de Bouwers zijn.** Gebruikers moeten denken in vragen, kansen en uitdagingen, de ontwikkelaar in oplossingen. Ontwikkelaars kunnen heel inventief zijn, terwijl gebruikers vaak denken "dat zal wel niet kunnen, dat is vast heel moeilijk of duur"! Als requirements engineer moet je er scherp op blijven dat de gebruiker, en jijzelf ook, niet in oplossingen gaan denken, centraal moet: wat is de vraag waar het hier om gaat? Welk probleem lossen we hier op? De ontwikkelaar mag daar een mooie oplossing voor verzinnen. Ieder zijn eigen werk toch?

Je kunt gerust stellen dat een requirements engineer als vak heeft om tussen twee, of soms meerdere werelden in te zitten. Zodat we kunnen doorvragen om de wensen van de gebruiker inzichtelijk te krijgen, zonder dat we met een zelfbedachte oplossing de ontwikkelaar beperken. En dat we in staat zijn om de prioriteiten en eisen vanuit het IT-perspectief zo te vertalen naar de gebruiker, dat hij zich goed bediend voelt.



**Linda Haak – van der Spek**

Linda.Haak@Ordina.nl

Linda werkt als Requirements Consultant bij Ordina. Zij traint en adviseert klanten op het gebied van requirements engineering, vaak in Agile omgevingen. Binnen Ordina traint en coacht zij collega's op het gebied van requirements engineering en Agile. Linda is trekker van themawerkgroepen binnen de afdeling Requirements, zo heeft zij bijvoorbeeld met collega's een intern Handboek Security voor Requirements Engineers geschreven. Linda is redacteur van het DREAMagazine.

Een requirements engineer vliegt zo steeds van de ene naar de andere wereld en heeft rekening te houden met de verschillende talen en culturen. Het is dan ook belangrijk om culturen, gebruiken, prioriteiten en talen te kennen. Communiceren, inlevingsvermogen en belangen behartigen zijn de kerncompetenties in dit werk. Als je die beheerst vlieg je in een vloeiende lijn heen en weer tussen de diverse werelden.