

Requirements vormen het hart van een IT-project. Ze vormen het contract tussen opdrachtgever en het projectteam. Bovendien krijgt iedere projectmedewerker er mee te maken. Toch hebben de meeste IT-projecten nauwelijks een strategie wat betreft requirements. Kunnen requirement management tools hierbij ondersteuning bieden? De auteurs van dit artikel onderzochten tools van IBM, Borland en Telelogic.



*bespreking*

# Requirement management

## *CaliberRM, Telelogic Doors, IBM RequisitePro*

Bij een grote autofabrikant was men bezig met het ontwikkelen van een nieuw autotype. Een van de requirements was als volgt: "Het chassis van de auto moet bestand zijn tegen een snelheid van 150 kilometer per uur achteruit." Een ijverige projectmedewerker dacht: "wat een onzin". Een andere requirement was namelijk dat de auto maximaal maar 30 kilometer per uur achteruit moet kunnen rijden. De medewerker schrapte daardoor het eerste requirement. De auto's werden geproduceerd met de 30 kilometer limiet, precies zoals in de requirements was opgegeven. De problemen kwamen toen men de nieuw geproduceerde auto's op de trein zette. Nu bleek pas waarom het geschrapte requirement ooit was opgenomen. Bij de auto's die achterstevoren op de trein stonden waren de achterraamen er uit gewaaid vanwege de hoge snelheid! Had het probleem met de auto's voorkomen kunnen worden met een requirement management tool? Voor we deze vraag kunnen beantwoorden moeten we eerst het begrip requirement management toelichten, en daarbij uitleggen hoe een tool hierbij kan helpen.

**WIJZIGINGEN** Volgens META Group is het mislukken van IT projecten in zo'n 60 tot 70 procent van de gevallen te wijten aan het slecht verzamelen, analyseren en managen van requirements. Daar tegenover staat dat uit onderzoek van de Standish Group is gebleken dat 50% van de redenen die gegeven worden voor het behalen van project succes direct gerelateerd waren aan het goed beheren van requirements. Dit zijn indrukwekkende cijfers, maar niet echt verbazingwekkend. Requirements vormen het hart van een IT-project. Het doel van de requirements is immers het vastleggen van het te bereiken resultaat voor gebruikers en project-sponsors. Ze vormen het contract tussen opdrachtgever en het projectteam. Bovendien krijgt iedere projectmedewerker er mee te maken. De analist legt de require-

ments vast. De ontwerpers vertalen de requirements naar een technisch ontwerp. De testers controleren of de requirements goed zijn geïmplementeerd en de projectleider bepaalt de planning en de voortgang van zijn project aan de hand van de requirements. Toch hebben de meeste IT-projecten nauwelijks een strategie wat betreft requirements.

In de meeste projecten worden de requirements in het begin van het project vastgelegd in documenten, die met een tekstverwerker opgesteld worden. Nadat de requirements zijn vastgelegd worden ze na een lang proces formeel goedgekeurd. Tot zover is er nog niet veel mis. Maar helaas, bijna altijd wijzigen de requirements gedurende het project. Dit kan verschillende oorzaken hebben: Er kunnen fouten ontdekt zijn in de requirements, en die moeten gecorrigeerd worden. Maar het is ook mogelijk dat de requirements na verloop van tijd verouderd zijn. Aanpassingen in de requirements worden vaak aangeleverd als wijzigingsvoorstel. Deze voorstellen kunnen via e-mailtjes verstuurd worden. Degene die de wijziging voorstelt zou dit eigenlijk moeten motiveren, maar dat is meer uitzondering dan regel. Op deze manier ontstaan vaak problematische situaties. Hoe hou je bij welke wijzigingen er allemaal op de originele requirements zijn? Hoe hou je bij dat het om een wijziging op een wijziging gaat? Wat is de impact van een wijziging op een requirement? Als we deze wijziging doorvoeren heeft het dan ook gevolgen voor andere requirements? Wie is er verantwoordelijk voor een requirement? Wie heeft de originele requirement ingediend?

We hebben zelf een project meegemaakt, waar de requirements bij elkaar meer dan 800 pagina's besloegen. Op het moment dat de wijzigingen kwamen was het onbegonnen werk om de impact hiervan te bepalen.

**STRATEGIE** Om dergelijke problemen het hoofd te kunnen bieden is het noodzakelijk om een goede requirements strategie te bepalen. De discipline die zich bezig houdt met het bepalen van een goede requirements-strategie noemen we requirement management. Door de hoeveelheid requirements en de complexiteit van alle requirements bij elkaar is het gebruik van een goede tool vaak onontbeerlijk. De meest belangrijke begrippen binnen requirement management zijn: requirements type, requirements attribuut, requirements hiërarchie en traceability.

Het requirements type geeft van wat voor type een requirement is. Dit kan bijvoorbeeld van het type use case zijn, maar ook een user requirement. Door requirement typen te definiëren kunnen soortgelijke requirements bij elkaar gegroepeerd worden. Elk requirement van hetzelfde type heeft ook dezelfde requirements attributen. Met behulp van attributen kan men eigenschappen van een requirement toekennen. Voorbeelden van een requirement zijn: de tekst (de eigenlijke requirements), maar ook het versienummer van de requirements, de iteratie waarin de requirement geïmplementeerd gaat worden, enzovoort.

Met requirement hiërarchie bedoelen we requirements van hetzelfde type, waarvan sommige een deel vormen van de in hiërarchie hoger liggende requirements. Een voorbeeld hiervan is een use case, waarbij de

stappen in de use case als aparte requirements worden opgenomen, maar wel een onderdeel zijn van de use case requirement zelf. Dit laatste kan gebeuren omdat bijvoorbeeld niet alle stappen in dezelfde iteratie worden geïmplementeerd. Een ander belangrijk aspect is het koppelen van verschillende requirements (van verschillende types) aan elkaar. Een wens van een gebruiker is bijvoorbeeld gekoppeld aan meerdere use cases en elke use case is op zijn beurt weer gekoppeld aan testcases en aan ontwerp. Door middel van traceability kan worden aangegeven welk requirement aan welk andere requirement gekoppeld is. Mocht nu in de toekomst ooit de gebruikerswens veranderen dan kan met behulp van traceability snel gekeken worden voor welke andere requirements het direct gevolg heeft. Dit maakt het bepalen van impact een stuk eenvoudiger.

Voor het ondersteunen van requirements bestaan verschillende tools. Wij hebben de drie meest gebruikte tools onderzocht. Het gaat om RequisitePro van IBM, DOORS van Telelogic en CaliberRM van Borland.

**REQUISITEPRO** De naam Rational is onlosmakelijk verbonden met het Rational Unified Process. Een methodiek om objectgeoriënteerde systeemontwikkeling op een iteratieve manier te doen. RequisitePro is niet alleen voor RUP-projecten geschikt, maar kan onafhankelijk van de methode worden ingezet. Wel wordt de gebruiker de mogelijkheid gegeven om een voorbeeld-database te gebruiken die al ingericht is voor RUP.

RequisitePro heeft van de onderzochte tools de beste integratie met het meest gebruikte tool voor het bijhouden van requirements: Microsoft Word. IBM combineert hiermee het beste van twee werelden: Word documenten voor het vastleggen van de requirements en een database voor het bijhouden van de attributen van die requirements. Bij het aanmaken van de requirement-tekst in Word zal RequisitePro een bijbehorende requirement aanmaken in de database. De requirement zal vanaf dat moment up-to-date worden gehouden: zodra de tekst in het Worddocument veranderd, kan deze eenvoudig worden doorgevoerd in Requisite. Dit geldt ook andersom: als de requirement in RequisitePro veranderd, kan deze in Word doorgevoerd worden.

**INTERNE DATABASE** Na de installatie van RequisitePro wordt in Word een speciaal menu toegevoegd om de koppeling met RequisitePro in stand te houden. Daardoor worden de requirements zowel in het document als in de tool bijgehouden. Het requirement document blijft dus bruikbaar, terwijl de voordelen van het gebruik van een requirement tool ook aanwezig zijn. De meeste tools kunnen wel Worddocumenten importeren (zie Figuur 1), maar kunnen deze het importeren niet meer up-to-date houden. Het document zal dus lang-



FIGUUR 1



FIGUUR 2

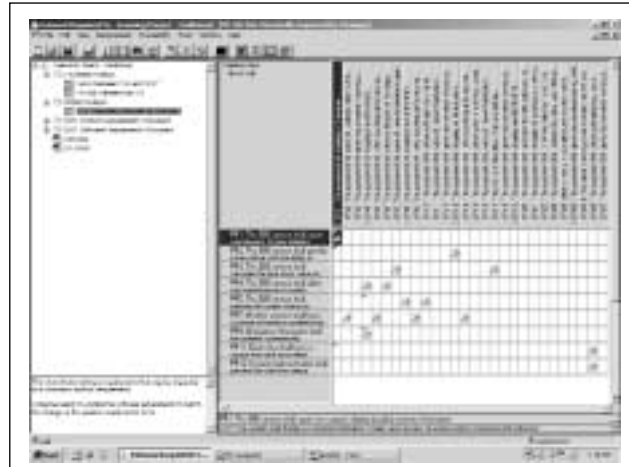
zaamaan verouderen. Een nadeel van deze aanpak is dat Word-documenten dan niet in een versiebeheersysteem gezet kunnen worden. RequisitePro heeft een interne database, die zijn eigen versiebeheer bijhoudt. Een database wordt over het algemeen niet in versiebeheer gezet. Bovendien wordt de koppeling met RequisitePro verbroken bij het ophalen van Worddocumenten uit versiebeheer. Overigens is het ook mogelijk om requirements te maken die alleen in de database voorkomen. Het is dus niet verplicht om ze in een document op te nemen.

Een ander nadeel van RequisitePro is dat er geen standaardrapportage ingebouwd is. Er kunnen wel afdrucken gemaakt worden van de views. Maar als er een specifiek rapport afgedrukt dient te worden, dan kan dat alleen door gebruik te maken van het programma Soda. Soda is een tool dat door Rational ontwikkeld is om uit de verschillende Rational-tools informatie te halen en deze via een Soda template te presenteren. Soda is een onderdeel van verschillende Rational Suites.

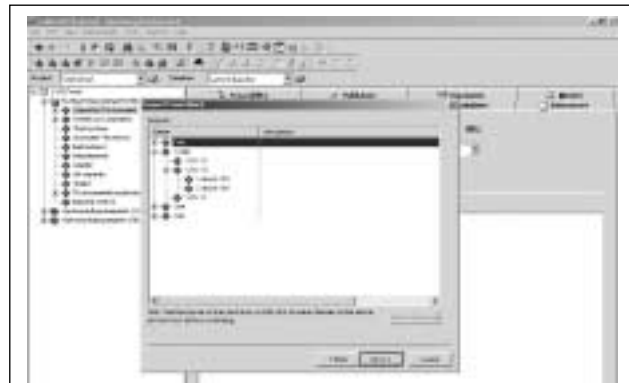
Het ontbreken van een compare functie is hinderlijk. De historie wordt wel per requirement bijgehouden, maar het is onbegonnen werk om met de hand verschillen te zoeken tussen groepen requirements. Ook het ontbreken van een baselinemechanisme is erg vervelend (als noodgreep de database in versiebeheer zetten gaat wat ver).

**DOCUMENTGERICHT** Met RequisitePro is het mogelijk om eenvoudige metrieken weer te geven over de requirements in het project. Deze metrieken worden in Excel als een grafiek weergegeven. Helaas zijn de mogelijkheden beperkt: Er wordt alleen aangegeven wat het totaal aantal requirements is, wat aan een bepaalde eigenschap voldoet (bijvoorbeeld het totaal aantal use cases waarvan de tekst gewijzigd is). Het zou beter zijn als opgezocht kan worden om welke requirements het gaat (van 20 use cases is de tekst gewijzigd, welke zijn dat dan?).

Een sterk punt van RequisitePro is de mogelijkheid om overzichten te maken van requirements. Voor deze overzichten kan uitgebreid aangegeven worden welke requirements getoond moeten worden. Zo kan bijvoorbeeld een overzicht getoond worden van alle requirements van een bepaald type (zie Figuur 2). Bijzonder is dat ook gefilterd kan worden op de traceability tussen requirements (zie Figuur 3). Dit biedt de mogelijkheid om te controleren of requirements compleet en consistent zijn. Normaal gesproken is dit een tijdrovend karwei. Door deze optie is zo'n controle veel sneller uit te voeren. Er kan bijvoorbeeld een lijst gemaakt worden van alle incomplete businessrequirements. In dit geval zijn dat requirements die geen (traceability-)relatie hebben met een onderliggende functionele requirement.



FIGUUR 3



FIGUUR 4

RequisitePro heeft de mogelijkheid om op elke view een filter te zetten. De filtermogelijkheden zijn redelijk uitgebreid, en de filters blijven automatisch bewaard. Samenvattend kunnen we stellen dat RequisitePro een sterk documentgerichte tool is. Hoewel de tool de requirements intern in een database opslaat, ligt de nadruk op de integratie met Worddocumenten.

**CALIBERRM** Dankzij de overname vorig jaar van Starbase, heeft Borland CaliberRM geacquireerd. CaliberRM is goed geïntegreerd met de andere Borland-tools. Borland zal deze integratie in de toekomst nog verder uitbreiden, waardoor het mogelijk wordt om vanuit andere tooling via de IDE de requirements direct te benaderen, zonder dat de gebruiker een nieuwe tool moet opstarten. Door gebruik te maken van een onderliggende DataMart is het mogelijk om informatie te verkrijgen per requirement gedurende alle fasen van de software-ontwikkeling. Een voorbeeld hiervan is het aantal testbevindingen dat bij een requirement hoort, voordat de implementatie van deze requirement goedgekeurd wordt.

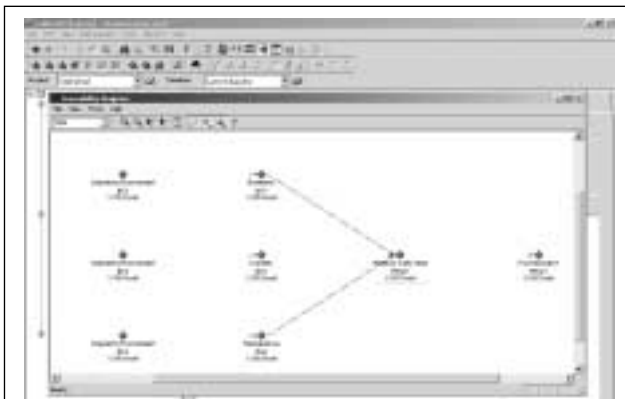
CaliberRM is ingedeeld met aan de linkerkant een boom met de requirements. Zodra een requirement geselecteerd is, wordt aan de rechterkant de informatie



FIGUUR 5

over deze requirement getoond in een aantal tabbladen. Een 'detail' tabblad bevat de belangrijkste gegevens over de requirement zelf. Denk hierbij aan naam, versie, status en beschrijving. Maar daarnaast zijn ondersteunde tabbladen opgenomen zoals: verwijzingen naar documenten, traceerbaarheid, historie en een discussiegroep. Deze laatst genoemde mogelijkheden zijn overigens in vrijwel elke requirement management tool aanwezig. CaliberRM biedt de mogelijkheid om zelf tabbladen aan te maken. Dit zorgt ervoor dat ook de zelf gedefinieerde attributen overzichtelijk blijven. In de meeste andere tools staan alle zelf gedefinieerde attributen op één scherm.

Als er bestaande requirements zijn in de vorm van een Worddocument, dan kunnen deze geïmporteerd worden met behulp van een wizard. Bij het importeren geeft de gebruiker aan hoe de requirements te herkennen zijn. Binnen een Worddocument zijn voor de requirements vaak een bepaalde stijl gebruikt, zoals bijvoorbeeld kopteksten. Een sterk punt van CaliberRM is dat bij het importeren de hiërarchie van de requirements meegenomen wordt (zie Figuur 4). Als een requirement in een document is opgesplitst in kleinere delen, dan worden bij het importeren deze requirements opgenomen als deel van de hoofdrequirement. Dit punt is belangrijk omdat sommige andere tools dit niet zo gracieus oplossen. Ondergetekende heeft eens tijdens een project met een ander tool geprobeerd om een document te importeren. Daarbij werden alle requirements



FIGUUR 6

op een hoop gegooid, waarna met drag en drop hiërarchie in de requirements aangebracht moesten worden. Zodra de requirements zijn ingevoerd, hetzij handmatig, hetzij door importeren, dan zullen deze regelmatig geraadpleegd en (hopelijk wat minder vaak) gewijzigd worden.

**GRID** Bij het raadplegen wordt vaak een bepaald deel van de requirements van belang zijn. Vrijwel altijd wordt er één type requirement tegelijk getoond (zie Figuur 5). Van deze requirements zal vaak maar een klein deel van belang zijn. Een gebruiker zal alleen dat deel van de requirements willen zien dat voor hem op dat moment van belang is. Zo worden er altijd maar een paar use cases tegelijk gebouwd. Het tonen van een deel van de requirements kan door middel van een filter. Met dit filter wordt aangegeven welk deel van de requirements getoond wordt, bijvoorbeeld alleen business requirements of alle requirements waarvan de prioriteit hoog is.

In CaliberRM kunnen de requirements in een grid getoond worden. In deze grid kan men met behulp van een filter een deel van de requirements te tonen. Het is mogelijk te filteren op requirements type of attribuut, en het mogen maximaal twee criteria zijn. Het filteren is op een onhandige manier ingebouwd. Zodra een bepaald filter is gekozen, wordt dit niet automatisch vastgehouden. (In requisite wordt dit een view genoemd). Als het grid wordt afgesloten, wordt de instelling van het filter niet bewaard. Bij een volgende keer openen verschijnen weer alle requirements zonder filter. Om het filter te bewaren moet het eerst expliciet opgeslagen worden in een bestand. Bij het openen van het grid moet dat bestand dan weer geopend worden. Dit is niet handig, vaak wil een gebruiker één filter instellen en daar een tijdje mee werken.

CaliberRM kent ook een traceability matrix. Een traceability matrix toont het verband tussen twee soorten requirements. Horizontaal wordt de ene requirement gezet, en verticaal staat de andere. In de vakjes van de matrix staat aangegeven welke requirements met elkaar in verband staan.

Ook hier komt de onhandigheid van het filteren naar boven. Het is standaard dat alle requirements zowel horizontaal als verticaal getoond worden. Hier moeten eerst filters gezet worden, die expliciet opgeslagen en geopend moeten worden.

**AANVULLING** Een unieke optie van CaliberRM is het traceability diagram. (zie Figuur 6). In dit diagram wordt net als in de traceability matrix het verband getoond tussen verschillende requirementstypen. Maar in het traceability diagram worden alle relaties voor één

bepaalde requirement getoond. Het is een diagram waarin door middel van pijlen de relaties worden weergegeven. Met dit diagram is het mogelijk de verbanden tussen alle verschillende requirementstypen te zien. Men kan de afhankelijkheid van het hoogste naar het laagste niveau volgen. Dus van business requirements naar functionele requirements en verder naar software requirements. Het traceability diagram toont niet alle relaties op alle niveaus, dan zou het diagram onleesbaar worden. Maar door op een requirement te dubbelklikken kan een andere requirement geselecteerd worden, waarna andere relaties zichtbaar worden. Al met al is het traceability diagram zeer nuttig, en vormt een goede aanvulling op de traceability matrix.

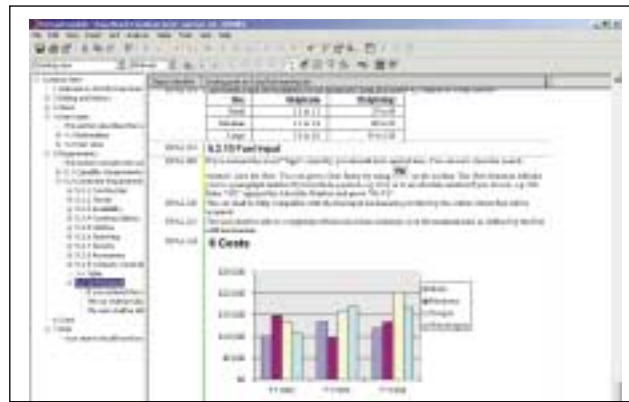
CaliberRM heeft verder de mogelijkheid om de traceability relatie te leggen tussen requirement in verschillende projecten. Dus als er twee projecten in CaliberRM zijn aangemaakt, is het mogelijk om een relatie aan te leggen tussen requirements uit beide projecten. Deze relatie wordt zichtbaar in het traceability diagram.

Requirements kunnen ook gedeeld worden tussen projecten. Dit betekent dat de beschrijving van een requirement op één plaats wordt vastgelegd. Meerdere requirements kunnen dan gebruik maken van deze beschrijving. Een toepassing hiervan zijn bijvoorbeeld standaards en normen die aan verschillende projecten worden opgelegd. Maar het kan ook zijn dat er sprake is van een productlijn, waarin de verschillende producten requirements kunnen hergebruiken.

Het in gedeelten bekijken van projecten is lastig, er kan wel gefilterd worden. Stel dat een gebruiker alleen hoofdrequirement 1 tot en met 4 wil zien, dan wil hij ook weten welke onderliggende requirements daarop betrekking hebben.

**DOORS** Telelogic – de kleinste van de drie, wat zeker niet aan de kwaliteit van het product af te lezen is - verkoopt DOORS niet als onderdeel van een suite. Het bedrijf is van mening dat een klant bewust voor requirement management en DOORS hoort te kiezen en niet voor een suite waar toevallig een requirement management tool bij zit. Ondanks deze strategie draagt Telelogic er zorg voor dat de integratie met hun andere tools zeer goed is.

Bij Telelogic wordt niet uitgegaan van een standaard proces. De gedachte daarbij is dat ieder bedrijf anders is en dus ook een ander ontwikkelproces zal hebben. Wel wil Telelogic klanten graag helpen bij het opzetten van een goede requirementsstrategie. Zo organiseert Telelogic regelmatig cursussen, die gegeven worden door Suzan Robertson. Zij geldt als een onafhankelijke expert op het gebied van requirement management.



FIGUUR 7

Tijdens de cursus worden de verschillende aspecten van requirement management uitgelegd, waarna de cursist zelf kan bepalen wat voor zijn bedrijf het beste is.

**TRAININGDATABASE** Bij het bekijken van Doors valt meteen op dat het de tekstuele opmaak van requirements ondersteunt. In de groep van door ons onderzochte tools was het daarmee een gunstige uitzondering. Door deze ondersteuning wordt het ook mogelijk om plaatjes op te nemen als een onderdeel van een requirement. Requirements behoeven niet zelden visuele ondersteuning, en daarom is het jammer dat de meeste requirement management tools hier niet mee om kunnen gaan. DOORS biedt de mogelijkheid om een screenshot daadwerkelijk bij een GUI-requirement te voegen. Als het screenshot verandert, blijft de oude versie in het versiebeheer van DOORS nog bestaan. DOORS biedt ook de mogelijkheid om OLE-objecten toe te voegen aan een requirement, bijvoorbeeld een Excel grafiek (zie Figuur 7).

Een handige manier om heel snel wegwijs te worden in DOORS is door gebruik te maken van de meegeleverde trainingdatabase. De trainingdatabase bevat een kort introductie van de meest gebruikte functies binnen DOORS. De training zelf is niet heel uitvoerig, maar geeft je wel een goede indruk van de meest gebruikelijke mogelijkheden van DOORS. Wil je echter meer weten en dieper op de verschillende functies van DOORS ingaan, heb je de mogelijkheid om een DOORS tooltraining bij Telelogic te volgen, of gebruik te maken van de uitvoerige documentatie die zowel digitaal als op papier meegeleverd wordt met de aanschaf van DOORS.

Het onderbrengen van al je requirements in een tool biedt vele voordelen, maar in elk project komt er een tijd dat je de requirements moet laten zien aan mensen die niet over dit tool beschikken. Gelukkig heeft DOORS hiervoor een zeer uitgebreide exportfunctie. Het is mogelijk om je requirements direct te exporteren naar onder meer Word, Excel, PowerPoint, Outlook,

**Advertentie**

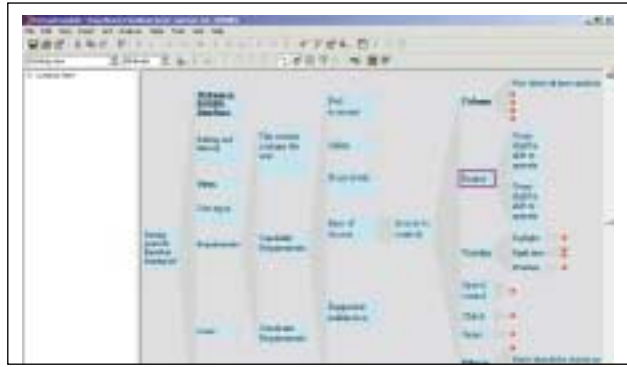
HTML, Rich Text Format en Microsoft Project. Vooral de koppeling tussen Microsoft Project en DOORS is goed doordacht en voor elk zichzelf respecterende projectleider een must.

**RECHTENSYSTEEM** Binnen DOORS worden requirements gegroepeerd in modules. Een module is een soort document waarin requirements staan die allemaal van hetzelfde requirementstype zijn. Heb je op een project acht verschillende requirementtypes dan heb je minimaal acht verschillende modules. De requirements in een module staan onder elkaar gerepresenteerd. De attributen die bij een requirement horen staan naast elkaar. Het is mogelijk om per module verschillende views te definiëren, waarin je aangeeft welke requirementattributen je wilt zien. De view biedt ook de mogelijkheid om een filter op de requirements te plaatsen, zodat je niet alle requirements te zien krijgt. Een speciaal type view van DOORS is de grafische weergave van de hiërarchie binnen een module (zie Figuur 8). Een minpuntje is het ontbreken van een gemakkelijke manier om grafisch de traceability te zien tussen verschillende requirements. Dit kan wel met tekst door een kolom toe te voegen.

DOORS heeft een zeer uitgebreid rechtensysteem. Elk object binnen DOORS kan zijn eigen rechten hebben of deze overerven van zijn parent. Zo kan in DOORS een requirement gemaakt worden, waar de rechten per attribuut verschillen. Zo kan het attribuut 'geplande uren' wel voor iedereen te lezen zijn, maar alleen aan te passen door de projectleider. Een andere attributen van deze requirement daarentegen kunnen door iedereen veranderd worden.

Eén van de meer geavanceerde functies van DOORS is het gebruik van DXL script. Met behulp van DXL script kunnen kolommen bijvoorbeeld berekend worden aan de hand van andere kolommen. Stel dat de projectleider een attribuut definieert waarin de planning in uren van een requirement staat, en dat hij de enige is die de inhoud hiervan mag aanpassen. Er is een ander attribuut waarin de reeds bestede uren staan van een requirement. Dan kan door middel van DXL script een derde attribuut gemaakt worden die de nog te besteden uren zelf berekend aan de inhoud van de twee voorgaande attributen (zie Figuur 9).

Doors bezit een redelijk uitgebreid rechtensysteem. Aangezien Doors met objecten werkt is het mogelijk om elk object andere rechten te geven. Hierdoor kan men bepaalde velden alleen openstellen voor bepaalde gebruikers. Stel dat de projectleider de enige is die bijvoorbeeld de geplande uren in mag vullen en andere team members dit wel mogen zien, maar niet wijzigen.



FIGUUR 8



FIGUUR 9

## CONCLUSIE

Onze belangrijkste conclusie is dat alle drie onderzochte tools voldoen aan de eisen die wij stellen voor het goed ondersteunen van requirement management. Het verschil zit hem vooral in de details. Wij vonden niet dat er één tool echt met kop en schouders boven de andere twee uitstak. Ons advies is dan ook de tool te kiezen, waar je al ervaring mee hebt of anders de tool te kiezen die van een leverancier is waarvan je al meerdere andere producten heb. Dit laatste is vooral ingegeven door het feit dat de tools het best integreren met producten die eigendom zijn van dezelfde leverancier.

Bij alle tools raden we aan om voor gebruik van de tool bij requirement management eerst goed na te denken over een requirement management strategie. Te vaak zien we projecten die wel een tool inzetten, maar niet duidelijk weten wat ze vast willen leggen. En dan maar hopen dat de tool hier mee helpt. Dit resulteert meestal in een berg geklaag over de requirement management tool en deze krijgt dan de onterechte kritiek dat ze niet goed werkt. Ook raden we aan - voor elk van de tools die wij onderzocht hebben - om in ieder geval de toolcursus te volgen, zodat je weet wat er allemaal met een goede requirement management tool mogelijk is.

*Dajo Breddels is werkzaam als Consultant Software Professionalisering bij Ordina SI&D Technology Consulting. Erwin Doreleijers is analist/ontwerper bij Ordina SI&D Requirements.*