

Over op modelgedreven

AFAS investeert 200 manjaar in ontkoppeling techniek

AFAS werkt aan een overstap van traditioneel handmatig programmeren naar modelgedreven applicatieontwikkeling. Daarmee wil het bedrijf de eigen slagvaardigheid vergroten en klanten flexibeler mogelijkheden bieden om het ERP-pakket AFAS Profit zo nodig naar hun eigen hand te zetten.

door: ROLF ZAAL / R.ZAAL@AUTOMATISERINGGIDS.NL **beeld:** DE BEELDREDAKTIE / RICK NEDERSTIGT

A FAS is het bedrijf achter Profit, één van de meest succesvolle ERP-producten in Nederland. Een dikke 10.000 klanten maken gebruik van dit pakket, waaronder accountantsbureau EY, copierfabriek Ricoh, postbedrijf Sandd, de NS-stations en de supermarkten Jumbo en Coop. Ook duizenden kleine en middelgrote bedrijven, non-profitorganisaties (waaronder 35 ziekenhuizen) en scholengemeenschappen gebruiken Profit.

In Leusden heeft het bedrijf honderd software-engineers aan het werk. Ze hebben hun handen vol aan zowel technische als functionele aanpassingen in de 8 miljoen regels code die bepalen wat Profit is en doet. Functioneel gaat het onder meer om nieuwe gebruiksmogelijkheden, gewijzigde wetten en veranderde cao's. Technisch betreft het vooral ondersteuning voor ICT-infrastructurele vernieuwingen, zoals moderne databases, tablets, private cloud en shared cloud. Klanten willen dergelijke aanpassingen liefst gisteren beschikbaar hebben. Toch zette AFAS anderhalf jaar geleden twintig van z'n beste software-engineers apart. Zij kregen de opdracht alles wat tot nu toe is gedaan nog eens over te doen, maar dan op een fundamenteel andere manier. Een manier die in de toekomst het tijdrovende programmeren in 3e-generatietalen als Visual Basic, C#, HTML en JavaScript overbodig moet maken. Gewerkt wordt aan een overstap op modelgebaseerd programmeren, waarbij nieuwe versies van Profit met een druk op de knop zijn te genereren op basis van modellen die op conceptueel niveau technisch en functioneel beschrijven wat die nieuwe versie (anders of meer dan de actuele versie) moet doen.

Ook voor klanten

AFAS' directeur Architecture & Innovation, Rolf de Jong, typeert de voorgenomen exercitie als 'het scheiden van techniek en functionaliteit'. Een scheiding die pas zinvol is als er ook een brug is. Die brug is nu onder constructie, in de vorm van een nieuw te ontwikkelen modelgebaseerde ontwikkelomgeving die in de AFAS-modelleertaal vastge-

legde modellen vertaalt in direct compileerbare 3GL-code. Dat vergroot de slagvaardigheid van AFAS als leverancier van software die bij de tijd is.

Interne efficiëntieverbetering is niet het enige doel dat AFAS met de transitie voor ogen heeft. De bedoeling is ook om de voordelen van modelgedreven software-engineering door te spelen aan de klanten. Steeds meer klanten passen AFAS' standaard-oplossing namelijk op een eigen wijze toe. Inmiddels gebruikt al omstreeks de helft van AFAS' klanten eigen maatwerk add-ons om Profit nauwer te laten aansluiten bij hun bedrijfsprocessen. Maar het ontwikkelen van dergelijke add-ons vergt technische kennis en is daardoor kostbaar, in geld en tijd. En met zekere regelmaat dwingen nieuwe versies van Profit tot het updaten van de add-ons. Beide problemen zijn in aanzienlijke mate te verlichten door, in plaats van add-ons te maken, op basis van aangepaste functionele modellen doodleuk een eigen versie van Profit te genereren. Daarbij is het overigens niet de bedoeling dat klanten Profit fundamenteel re-engineeren. De Jong: "Ons idee is klanten een standaard kernapplicatie te leveren, met daarbovenop aanpasbare best practices die men naar believen kan veranderen of uitbreiden, zonder dat dat risico's met zich meebrengt ten aanzien van zaken als beveiliging, betrouwbaarheid en onderhoudbaarheid."

De beslissing om voor klanten een zekere mate van modelgedreven re-engineering van Profit mogelijk te gaan maken, impliceert dat AFAS naast de uitvoerbare objectcodes van de kernapplicatie en de best practices, ook de modellen van best practices en de modelgedreven ontwikkelomgeving aan z'n klanten beschikbaar zal stellen. Dat doorleveren aan de eigen klanten kan licentie- en kostentechnisch lastig zijn als het zou gaan om een commercieel beschikbare ontwikkelomgeving van een derde partij, zoals BeInformed, Pegasys-systems of Mendix. Maar dat probleem is voor AFAS niet aan de orde, daar het bedrijf er voor kiest een eigen modelgedreven ontwikkelomgeving te ontwikkelen en geïntegreerd onderdeel te maken van de Profit-applicatie zelf.

Personele gevolgen

Inmiddels groeit het ontwikkelteam Next naar 30 man, terwijl er onverminderd geïnvesteerd wordt in het huidige AFAS Profit. Tientallen hooggekwalificeerde medewerkers voor een langdurig project onttrekken aan een bestaand software-ontwikkelteam, was volgens De Jong alleen mogelijk dankzij het feit dat de huidige 3GL-applicatiecode goed onderhoudbaar in elkaar steekt. Tegelijkertijd moet de doelstelling van het Next-project (drastische reductie van de uiteindelijk voor applicatiemanagement benodigde hoeveelheid werk) voor de achterblijvende software-engineers geen opwekkend vooruitzicht zijn. Maar volgens Rolf de Jong valt dat in de praktijk enorm mee. "Om te beginnen hebben we hier vanaf het begin in 2009 steeds open gecommuniceerd met het voltallige team. Iedereen begrijpt waarom we dit doen en wat op termijn de impact ervan is. De meesten zullen daar qua scholing en carrière op anticiperen, maar voorlopig is er nog jaren werk in de 3GL-omgeving. Combineer dat gegeven met de ambities van medewerkers om zich verder te ontwikkelen en dan weet je dat we waarschijnlijk geen probleem zullen hebben om de transitie wat mensen en loopbanen betreft in goede banen te leiden. Bovendien zullen er in de architectuurlaag en de generator van de nieuwe modelgedreven ontwikkelomgeving steeds vernieuwingen gerealiseerd worden in 3GL."

Vooralsnog is AFAS's eerste prioriteit dan ook niet afvloeiing, maar werving van software-engineers. AFAS werkt intensief samen met de Universiteit van Utrecht om deze ontwikkelingen met studenten te delen en nog onlangs plaatste het bedrijf een videoboodschap op internet waarin CEO Bas van der Veldt bèta's oproept te solliciteren. Om jonge software-engineers te inspireren startte AFAS op dev.afas.nl een platform over technologische ontwikkelingen op het gebied van softwareontwikkeling en evenementen, zoals het IT Talent Debate en de Meetup: Mobile .NET Developers.

software-engineering

en functionaliteit



Complexiteit bestrijden

Naast de wens om ook klanten modelgedreven aan Profit te laten sleutelen speelde echter nog een andere, zo mogelijk nog zwaarder wegende overweging om een eigen modelgedreven ontwikkelomgeving te ontwikkelen. Commerciële modelgedreven applicatieontwikkelomgevingen zijn doorgaans tamelijk generiek. De gedachte van de leverancier is immers iets te leveren dat zo algemeen mogelijk bruikbaar is. Maar met die rijkdom aan mogelijkheden sluipt ook complexiteit binnen. “Om dergelijke tools te gebruiken moet je toch verstand hebben van software-engineering, en dat is nu juist wat wij niet willen. Wij willen gebruikers in staat stellen volledig vanuit hun materiekennis aanpassingen te doen”, legt De Jong uit. De transitie van de huidige op derde generatietalen gebaseerde softwareontwikkeling naar een toekomstige software-factory op basis van functionele modellen en codegeneratie, wordt door AFAS conceptueel opgedeeld in 4 deel-uitdagingen:

1. Een taakgerichte gebruikersinterface die geschikt is voor diverse platformen.
2. Een cloudgebaseerde technische architectuur waarin beveiliging, schaalbaarheid en beschikbaarheid applicatiebreed wordt gemanaged, zodat degene die modelleert deze basisvoorwaarden niet meer kan verstoren.
3. Een modelleeromgeving waarin de volledige functionaliteit van Profit te beschrijven is, inclusief de denkbare klantspecifieke uitbreidingen en aanpassingen. Om het voor ontwikkelaars en klanten niet nodeloos gecompliceerd te maken kiest AFAS hier voor een eigen domein specifieke AFAS-modelleertaal, die een steile leercurve mogelijk maakt. Bij deze modelleeromgeving hoort uiteraard ook de eigenlijke codegenerator, die modellen vertaalt naar compileerbare 3GL.
4. Het uiteindelijke model dat de functionaliteit (kernapplicatie en best practices) van het huidige Profit beschrijft.

Vertrekpunt voor de ontwikkeling van deze modelleeromgeving is het huidige ERP-pakket Profit. De Jong: “Omdat we het beheer van de applicatie voor de gebruiker herkenbaar en eenvoudig willen houden, wordt de modelleeromgeving volledig in de applicatie geïntegreerd. Er blijken veel overeenkomsten met de administratieve vastlegging en we kunnen zo wederzijds alle voordelen van de technische-architectuur gebruiken.”

Hoewel bovenstaande opsomming chronologisch oogt, pakt het Leusdense ‘Next-team’ alle vier de deeluitedagingen gelijktijdig aan, waarbij tussenresultaten doorlopend aan elkaar worden getoetst. Zo wordt bijvoorbeeld de te ontwikkelen modelleeromgeving, dagelijks getest op z’n vermogen om uit de voeten te kunnen met het gestaag groeiende model van het nieuwe Profit.

Eens maar nooit weer

Voor het hele project heeft AFAS 150 à 200 manjaar begroot. Daarvan is inmiddels zo’n 80 manjaar besteed. Een harde deadline voor de oplevering is niet vastgelegd; dit onder meer om te voorkomen dat deadlinestress de inhoudelijke focus van het ontwikkelteam zou kunnen verstoren. Om diezelfde reden wordt het project met de codenaam Next aangeduid, en niet met een voorgenoemen versienummer. Het is immers absoluut niet duidelijk hoeveel conventionele versies van Profit nog zullen worden uitgebracht vooraleer versie-Next klaar is.

Dat de 3GL-ontwikkeling niet stilstaat terwijl het Next-team de aan de modelgedreven toekomst spijkert, betekent dat alle aanpassingen die in de huidige 3GL-werkelijkheid plaatsvinden, ook weer in de ‘rijdende Next-trein’ moeten worden ingebracht. Niet alleen functioneel, wat nu dus relatief simpel en snel gaat, maar ook wat de achterkant betreft. Veel dubbel werk, dus? De Jong: “Ja, maar het kan niet echt anders. Dat accepteren we, omdat we weten dat we het hierna nooit meer hoeven te doen.”

ROLF DE JONG:
‘WE WILLEN
ONZE KLANTEN
EEN STANDAARD
KERNAPPLICATIE
LEVEREN, MET
DAARBOVENOP
AANPASBARE BEST
PRACTICES’

