



[We Love IT](#) > [Magazine](#) > [Inhoud - 2007 uitgave 2](#) > VX Company

Tags:



Applicatie ontwikkeling nu (met Java/J2EE) en vroeger (met Cobol)

De huidige Java-professionals zijn jong, wild en opgegroeid met PC's waarop ze alleen werken. Hoe anders was dat in de tijd van Cobol. Duizenden gebruikers werkten gelijktijdig op één systeem. Dat stelde hoge eisen aan het computerwerk in die tijd. Die ervaring bracht de Cobol-generatie veel wijsheid. Tegenwoordig wordt vanaf PC's via webservices weer op grote systemen gewerkt. Maar de Java-professionals hebben de Cobol-leerschool niet doorlopen. In dit artikel staan we stil bij de evolutie van Cobol naar Java. En we halen het stof van de oude IT-lesse. Ter leering natuurlijk, maar ook ter "vermaeck".

Wijsheid van traditionele aanpakken

De Cobol standaarden zouden toch standaard onderdeel moeten zijn van cursusmateriaal. Maar Java-professionals slaan die klassieke leerschool over. Met alle gevolgen van dien. Ze lijken te vergeten dat een project ook een keer afgerond moet worden. Iets opleveren is er nauwelijks bij, laat staan dat er uitspraak gedaan wordt over wat wanneer wordt opgeleverd. Dat wordt verbloemd door het bedenken van nieuwe methoden zoals eXtreme Programming (XP). Zo kunnen ze ook formeel zeggen dat ze echt niet weten wat ze uiteindelijk zullen opleveren. En worden de Javanen wel gedwongen zich te committeren aan een oplevering, dan gaan stress, zweet, liters cola en koffie en kilo's pizza daaraan vooraf. Dit leidt tot een ongezonde situatie! Het wordt tijd dat professionals getraind worden in de nieuwe Java ontwikkelomgevingen met de wijsheid van de traditionele aanpakken.

Makkelijk omspringen met grote gegevensbestanden

En zo'n traditionele aanpak is Cobol. Cobol is de afkorting van *COmmon Business Oriented Language* (algemeen zakelijk georiënteerde taal). Deze programmeertaal stamt uit de jaren 60 van de vorige eeuw. Cobol is een taal die gemakkelijker te lezen, te schrijven en te onderhouden is. Het was een ware verademing voor die tijd waarin Assembly talen vrij gangbaar waren. Cobol is gericht op het makkelijk omspringen met grote gegevensbestanden in een vaak batch-georiënteerde omgeving. De jaren zeventig werden gedomineerd door de 3GL's (met name Cobol), de jaren tachtig door de 4GL's en de afgelopen tien jaar voerden OO-talen (zoals Java) de boventoon. Tegenwoordig heeft Cobol een slecht imago en zijn de IT-professionals van vandaag niet uitgerust met Cobol kennis. Het web is belangrijker, dus programmeren voor het web ook.

Objecten en relaties

Een mooi boek is "An introduction to Database Systems" van meneer Chris Date (hij is een van de grondleggers van relationele database). Bij de jongere Java professionals niet echt bekend, maar voor de ouderen onder ons een icoon. Het relationele model is een manier om data te structureren en is ook declaratief. We geven niet aan welke stappen doorlopen moeten worden, maar alleen wat we willen weten. Het wat dus en niet het hoe. Hierdoor kan goed geoptimaliseerd worden. De optimizer in de database heeft alle vrijheid om de meest efficiënte oplossing te zoeken. Dit alles heeft zijn weerslag als je het wilt combineren met object technologie. Bijna alle object-relational tools vertalen een tabel naar een klasse. En dat is wat anders. Klassen hebben geen unieke sleutels maar een magische eigen identiteit die alleen met pointers (references) onderhouden kunnen worden. Ze zijn gebaseerd op inkapseling en laten liever niet alle attributes zichtbaar voor de wereld. Het is niet declaratief. Feitelijk is een klassemodel met collections een stap terug naar de oude netwerk database (Codasyl model), een toonaangevende techniek in 1981.

Inkapseling keert zich tegen ons. Alle nadelen van de netwerk databases (bijvoorbeeld niet declaratief en fysieke toegangspaden die in de code sluipen) kunnen ook optreden bij klassemodellen, waardoor het ontwerp star wordt. Ook zaken zoals caching, locking en sharing moet je zelf maar via allerlei slimme patterns regelen, terwijl databases hier zulke mooie volwassen techniek voor hebben. Het lijkt een stap terug! We gooien het oude boek van Chris Date nog maar niet weg.

Java is terug naar af

De eerste Java/J2EE-specificaties dateren van 1998 (bèta). De ontwikkeling van Java startte in 1991 en werd in 1995 voor het eerst publiekelijk aangekondigd. Hoewel we nu een flink aantal jaren verder zijn, is de snelheid waarmee Java/ J2EE applicaties ontwikkeld worden nog steeds bedroevend laag. De focus ligt maar al te vaak op de techniek (b.v. raamwerken, die maar moeizaam met elkaar te koppelen zijn), in plaats van op de business. Het classificeren van business rules vindt maar al te weinig plaats. Vanwege die teleurstellende progressie, grijpen organisaties voorzichtig terug naar de 4GL ontwikkelproducten van het vorige decennium. Sommige leveranciers van deze oude bekenden melden weer omzetgroei! De komst van 4GL leverde immers een productiviteitsverbetering van minimaal een factor 3 (gemeten ten opzichte van Cobol). Met de komst van Java/J2EE lijken we deze productiviteitsverbetering weer in een klap te verliezen. We kunnen stellen dat Java/J2EE een

goede basis is. Nu gaan we op zoek naar de oplossing voor de productiviteit. Gewoon om weer net zo snel te ontwikkelen als in de 4GL tijd.

Betere productiviteit met Oracle Java

Oracle heeft breed ingezet op Java en biedt met JDeveloper een van de meest productieve Java ontwikkelomgevingen. Kenmerkend hierbij is Oracle's brede ondersteuning voor de Java standaarden, goede ondersteuning voor bewezen open source tools en libraries en uiteraard Oracle's Application Development Framework (ADF). ADF stelt ontwikkelaars in staat om productiever te ontwikkelen eventueel in combinatie met bijvoorbeeld aan JSP, JSF en EJB 3 of Toplink. Als de businesscase daarom vraagt kan gebruik worden gemaakt van Oracle JHeadstart. Door de declaratieve manier van specificeren worden verschillende ADF componenten gegenereerd, wat een enorme productiviteitswinst oplevert. Vooral voor Java-ontwikkelaars met een Oracle Designer achtergrond is Oracle Jdeveloper/ Jheadstart een perfecte combinatie van Java en het Oracle platform.

Betere productiviteit met open source

De open source community heeft de afgelopen jaren een enorme bijdrage geleverd aan de de productiviteit van software ontwikkeling. Het Spring framework loopt hier voorop en levert naast meer productiviteit, ook betere kwaliteit. De behoefte aan zware enterprise oplossingen is niet altijd aanwezig. De markt vraagt steeds vaker om lightweight oplossingen. Daar biedt Spring in combinatie met Hibernate en Struts, JSF of Tapestry een uitermate geschikte oplossing voor. Sleutelwoorden bij deze oplossingen zijn Domain Driven Development en AOP (Aspect Oriented Programming). Oplossingen die uitermate goed samengaan met Agile methodes zoals XP of Scrum.

IT professional ben jij Java ready?

Als Java professional begeef je je in een speelveld van heel veel ontwikkelingen. Uiteraard misstaat enige basiskennis van het relationele model hierbij niet (het boek van Chris Date). Maar ook professionele vaardigheden zoals plannen en organiseren worden steeds belangrijker. Verder moet je minstens een Java/J2EE omgeving van een leverancier (b.v. Oracle) goed beheersen en ben je bekend met een behoorlijk aantal open source producten. Gelukkig worden veel open source initiatieven door de productleverancier toegejuicht en opgenomen in de verschillende Suites. Als Java professional ben je minimaal gecertificeerd SCJP en als Java Architect heb je uiteraard meerdere Sun certificaten. Kennis en ervaring van producten zoals Oracle Java/J2EE Suite en de open source gerichte MyEclipse Suite worden steeds vaker gevraagd. We zien deze producten dan ook de komende periode steeds vaker terug bij organisaties en hopelijk zijn de meeste valkuilen nu getackeld.

Maikel Alderhout is technisch manager bij VX Company. VX Company is een IT-dienstverlener voor ontwikkeling, integratie, testen en beheer van bedrijfskritische infrastructures en applicaties. Dit doen wij met Microsoft .NET, Java/J2EE, Oracle en Unix. Het bedrijf telt 250 medewerkers, bestaat sinds 1988 en is gevestigd in Baarn.