

donderdag 9 januari 2003

Ervaring met objectoriëntatie en frameworks beslissend voor succes J2EE

Door: Patrick van Driel

Verschenen in Automatisering Gids nr. 2, 2003

Ondanks de populariteit van Java 2 Enterprise Edition hebben nog maar weinig organisaties ervaring met dit platform voor systeemontwikkeling. Voor veel automatiseerders is de omschakeling groot. Met als gevolg dat er regelmatig iets misgaat. Patrick van Driel belicht de technische en organisatorische problemen die men moet oplossen om van J2EE een productieve ontwikkelingsomgeving te maken.

In de afgelopen jaren hebben veel (ICT-)organisaties gekozen voor systeemontwikkeling op basis van internettechnologie. Voor de ontwikkeling van grootschalige bedrijfskritische toepassingen bijvoorbeeld systemen voor bankieren via internet wordt vaak J2EE (Java 2 Enterprise Edition) als uitgangspunt genomen. Waarom? De specificaties van J2EE zijn gericht op applicaties die grote aantallen transacties kunnen verwerken met korte responsetijden en een hoge beschikbaarheid en betrouwbaarheid. In tegenstelling tot het belangrijkste alternatief .Net van Microsoft, draaien Java-applicaties op vrijwel alle platformen die in grote organisaties voorkomen (naast Unix en Linuxvarianten ook op een mainframe, bijvoorbeeld onder OS/390 van IBM). Bovendien was J2EE enkele jaren eerder dan .Net beschikbaar voor ontwikkelaars en heeft het daardoor inmiddels een aanzienlijk marktaandeel verworven en zijn de kinderziekten verdwenen.

Desalniettemin hebben veel organisaties nog weinig ervaring met systeemontwikkeling op basis van J2EE. Omdat deze standaard omvangrijk is en productversies van leveranciers elkaar snel opvolgen, is de implementatie in ICT-organisaties buitengewoon lastig en voorbehouden aan specialisten. Het voor J2EE kenmerkende werken en denken in objecten en componenten vormt een grote omschakeling voor veel (traditionele) automatiseerders. Het gevolg is dat J2EE-projecten vaak met vallen en opstaan verlopen.

Na enkele 'experimenten' op technologisch vlak komt er een moment waarop men zich afvraagt: hoe verhogen we de productiviteit van de J2EE-ontwikkelingsactiviteiten? Hoe krijgen we de productiviteit op minimaal hetzelfde niveau als van meer traditionele technologieën zoals Cobol? Hoe verdienen we de gedane investeringen terug? Dergelijk kostenbewustzijn wordt niet alleen gestimuleerd door de huidige (neergaande) economische ontwikkelingen, maar vooral ook door het besef dat de behoeften van de klant (ook van een interne klant) altijd bepalend zijn voor het ontwerp van de oplossing. De hamvraag is: hoe krijgen en houden we de productiviteit van J2EE-ontwikkelingsprojecten hoog? Anders gezegd: Hoe kan met J2EE-technologie zoveel mogelijk functionaliteit worden gerealiseerd tegen zo laag mogelijke kosten? En vanzelfsprekend met een hoog kwaliteitsniveau en in een korte doorlooptijd.

Stappen

Naast de keuze voor J2EE als standaard voor (nieuwbouw)systeemontwikkelingsprojecten kiest een organisatie meestal ook een standaard voor de ontwikkelingshulpmiddelen en voor de runtimeomgeving van J2EE-applicaties. Te denken valt aan producten van commerciële leveranciers zoals IBM, Sun en BEA, maar ook aan vrij beschikbare opensourceproducten zoals Tomcat of JBoss.

De adoptie van nieuwe producten en een nieuwe standaard, zoals J2EE, door een ICT-organisatie verloopt veelal in de volgende stappen (niet noodzakelijk in deze volgorde):

-In het kader van een pilotproject wordt een 'proof of concept' ontwikkeld.

- Externe experts met ervaring met de nieuwe technologie, worden ingehuurd.
- Er worden standaarden opgesteld voor ontwerpers en programmeurs.
- Er worden generieke, herbruikbare componenten ontwikkeld zoals een objectgeoriënteerd raamwerk (framework) en een bibliotheek van utilities, bijvoorbeeld datumconversies of de '11-proef' voor bankrekeningnummers.
- De softwareomgeving en het beheer ervan, inclusief versiebeheer, wordt ingericht.
- Logistieke processen en hulpmiddelen voor het uitrollen van applicaties naar test- en productieomgevingen worden ingericht.
- Er worden 'echte projecten' uitgevoerd die productierijpe applicaties opleveren.
- Medewerkers worden opgeleid en gecertificeerd.

Het is duidelijk dat deze stappen niet specifiek zijn voor de adoptie van J2EE-technologie. Alle of in ieder geval de meeste van deze activiteiten heeft men in grote organisaties reeds eerder uitgevoerd voor andere nieuwe technologieën. Wat wel specifiek voor J2EE is en bovendien uitermate belangrijk voor de productieve inzet van Java-technologie, is ervaring met objectoriëntatie en de inzet van frameworks (raamwerken). Frameworks zijn een bekend fenomeen in objectgeoriënteerde omgevingen. In de context van J2EE bestaat zo'n raamwerk uit een set Java-componenten die in samenhang de interne technische werking van een applicatie bepalen. De functionele werking van een applicatie wordt per project als uitbreiding van een dergelijk raamwerk geprogrammeerd.

Een framework draagt bij aan de standaardisering en het hergebruik van architectuur, ontwerpen en software. Een goed framework maakt gebruik van zogenaamde patterns (patronen). Dit zijn gestandaardiseerde, in de praktijk bewezen oplossingen voor steeds terugkerende, vergelijkbare problemen. Een goed ontworpen framework, dat bovendien gebruikmaakt van (door anderen bedachte en getoetste) patterns, leidt tot verhoging van productiviteit en verlaging van kosten, zowel voor applicatieontwikkeling als voor applicatiebeheer. Het is echter niet gemakkelijk om een goed framework te ontwerpen en te implementeren in de organisatie. Gebrek aan kennis en ervaring met objectoriëntatie en patterns is een serieuze belemmering voor de productieve inzet van Java en J2EE. Zonder in details te treden, wordt hier een overzicht gegeven van de belangrijkste redenen waarom de technische en organisatorische invulling van frameworks in de praktijk zeer lastig is:

- Het ontwerpen van een framework is zeer lastig: het moet niet te algemeen zijn, maar ook niet te specifiek.
- J2EE-ervaring is schaars, dus duur, terwijl de kosten vaak juist omlaag moeten.
- De technologie, de tools en de standaarden zijn nog niet stabiel.
- De technologie heeft zich nog niet volledig bewezen voor grote bedrijfskritische toepassingen, bijvoorbeeld wat betreft performance, stabiliteit en beschikbaarheid.
- Men moet bedacht zijn op het gevaar van een te grote afhankelijkheid van één of meer leveranciers; het is raadzaam geen leveranciersspecifieke toevoegingen aan frameworks of producten toe te staan.
- Het is niet makkelijk om budgetten te krijgen voor standaardisatie en centrale coördinatie van ontwikkelings- en beheerprocessen, omdat het effect voor de business niet altijd direct aantoonbaar is.

Referentie

Elk framework is van twijfelachtige waarde als er geen duidelijke en gedetailleerde beschrijving is van hoe het gebruikt dient te worden. Eén van de belangrijkste faalfactoren is het ontbreken van een goede referentietoepassing. Dit is een complete en volledig functionerende toepassing die alles doet wat een gemiddelde administratieve webtoepassing doet, van het benaderen van een back-enddatabase tot en met de gebruikersinteractie op een

webpagina. Indien de ontwerpers van een framework hetzij een externe leverancier hetzij een intern project of afdeling geen referentieapplicatie beschikbaar stellen, hoe kan men dan verwachten dat ontwikkelaars in een project het betreffende framework op een productieve manier gebruiken?

In de praktijk komt het voor dat organisaties trachten één allesomvattend framework te realiseren waarin alle wensen en eisen van de organisatie zijn ingevuld. Die aanpak schiet z'n doel met betrekking tot productiviteitsverhoging voorbij. Een dergelijk framework is nooit af, wordt achterhaald door marktstandaarden en de investeringen in tijd en geld worden hoogstwaarschijnlijk niet meer terugverdiend. Uit productiviteitsoogpunt is één allesomvattend framework uitermate onverstandig. Uit technisch oogpunt gezien is het verleidelijk een dergelijk raamwerk te ontwikkelen, maar het zal leiden tot een onoverzichtelijk en complex geheel dat daardoor niet bijdraagt aan een verhoging van de productiviteit.

Het is belangrijk in te zien dat voor frameworks niet geldt 'one size fits all'. Qua functionaliteit is het vaak mogelijk onderscheid te maken tussen meer en minder complexe applicaties. Hetzelfde geldt voor de benodigde technologie.

Grote organisaties bouwen vele typen toepassingen voor internet, intranet en extranet. Aan die verschillende typen worden ook verschillende eisen gesteld. In de praktijk kan het merendeel van de te realiseren toepassingen gebaseerd worden op een relatief eenvoudig framework. Dit geldt bijvoorbeeld voor vrijwel alle administratieve intranettoepassingen waar de benodigde capaciteit en schaalbaarheid beperkt blijven. In overige gevallen, waar eisen gelden zoals een zeer hoge beschikbaarheid, een extreem piekgebruik of het benaderen van meerdere back-endsystemen binnen één transactie, is het wenselijk om meer geavanceerde (en daardoor relatief complexe) frameworks toe te passen, bijvoorbeeld gebruikmakend van Enterprise Java Beans.

Een organisatie dient per type toepassing vast te stellen welk framework het best past bij de eisen die aan de toepassing worden gesteld. Dus niet 'one size fits all', maar 'het juiste gereedschap voor de juiste klus'.

Productiviteit

Een systeemontwikkelaar behaalt een bepaalde productiviteit, dat wil zeggen, hij kan een bepaalde hoeveelheid functionaliteit per tijdseenheid realiseren. In het algemeen kan gezegd worden dat deze productiviteit afneemt naarmate de functionaliteit van een toepassing complexer wordt. Tevens geldt dat deze productiviteit afneemt naarmate de technische complexiteit van het framework dat hij gebruikt toeneemt.

Indien men kiest voor een allesomvattend framework leidt dit veelal onnodig tot veel productiviteitsverlies en frustraties bij ontwikkelaars en bij beheerders van de test- en productieomgeving. Een organisatie dient per type toepassing vast te stellen welk framework het best past bij de eisen die aan de toepassing worden gesteld. Binnen grote organisaties kunnen de meeste applicaties worden gecategoriseerd in een paar typen. Bij de ontwikkeling van kleine en eenvoudige toepassingen is het mogelijk om de onderliggende complexiteit af te schermen voor de systeemontwikkelaars. Bij de ontwikkeling van grootschalige, bedrijfsbrede systemen is het nu eenmaal een feit dat daar veel zaken bij komen kijken en het dus een complexe taak is. Het is een utopie te denken dat deze complexiteit volledig weggenomen kan worden door gebruik te maken van een framework of een generator.

Het is zaak de complexiteit te structureren en op die wijze behapbaar te maken. Per onderdeel kunnen dan technische details tot een bepaald niveau worden afgeschermd voor de ontwikkelaars. De ontwikkelaars moeten te allen tijde in staat zijn in objecten en componenten te denken (hetgeen op zich al een uitdaging is), maar zij moeten nooit de vraag voorgelegd krijgen een relatief eenvoudig probleem op te lossen met een complex framework.

Dit leidt onherroepelijk tot een lage productiviteit.

*Ir. Patrick van Driel is medeoprichter en directeur van Quinity BV te Utrecht
(patrick.van.driel@quinity.com).*