

Wij Java/J2EE-ontwikkelaars zijn natuurlijk trots op hun manier van software bouwen. Vooral de ervaren J2EE ontwikkelaars zijn trots op al die kennis en ervaring, bloed, zweet en tranen die zijn gaan zitten in het onder de knie krijgen van J2EE. Het is taaië materie, en het wordt met de dag uitgebreider, ingewikkelder en onoverzichtelijker.

# Spring en de zoektocht naar eenvoud

Natuurlijk wordt er goed over nagedacht en zijn er ook vereenvoudigingen, soms zelfs ingrijpende vereenvoudigingen. Maar, zolang nog niet alles naar wens is, is men geneigd J2EE steeds groter te maken. J2EE dijt uit, dikker en dikker. Meer XML, meer interfaces, meer regeltjes, meer tools, meer, meer, meer. J2EE-ontwikkelaars worden steeds trotser dat ze nog steeds in staat zijn om met al die ingewikkelde materie systemen te bouwen. Het wordt zo groot dat het onmogelijk incorrect kan zijn, toch?

Met een collega had ik het laatst over succesvolle software. Welke software wordt succesvol en welke software zal falen? Uit een zoekgeraakt artikel wist hij een aantal criteria voor succesvolle software te reproduceren. "Succesvolle software is toegankelijk", herinnerde hij zich. Succesvolle software is software die goed met 'fouten' om kan gaan. Browsers zijn een voorbeeld van succesvolle software die inderdaad met foute content kan omgaan. Mathematisch correcte, volledig validerende, onvergevende systemen als XML zouden volgens de auteur niet succesvol horen te zijn. Aan de andere kant zou leesbaarheid, ongeveer zoals die van XML, weer een belangrijke plus zijn. Het belangrijkste criterium dat hij zich kon her-

inneren was dat succesvolle software eenvoudig moet zijn, eenvoudig op meerdere niveaus. Eenvoudig in het gebruik en eenvoudig ontwikkeld. Pure eenvoud.

Als voorbeeld van eenvoudige software bracht hij Spring naar voren. Hij vertelde dat een J2EE-ontwikkelaar, die voor het eerst Spring ziet, direct "Zo eenvoudig kan het toch niet zijn?" uitroept. Nu probeer ik zoveel mogelijk bij te blijven in de J2EE-wereld, maar Spring had ik nog niet bekeken. Spring stond ergens boven aan mijn lijstje met nog uit te zoeken frameworks. Tja, je kunt niet altijd voorop lopen. Nu had ik wel eens wat gelezen over Spring maar ik had het nog nooit gebruikt. Een paar dagen na de succesvolle-software-discussie sprak ik een vriend van me, een echte J2EE-deskundige. Toen we het over Spring hadden zei hij: "Het eerste wat ik dacht was: Zo eenvoudig kan het toch niet zijn?" Het wordt duidelijk tijd dat ik me eens verdiep in Spring.

Voor de duidelijkheid, wat is Spring? Spring is een open source framework waarmee op een heel eenvoudige manier de meeste web- en fat client-applicaties gebouwd kunnen worden. Hoe eenvoudig is Spring

ten opzichte van J2EE? Je hoeft geen EJB-interfaces meer te maken, je hebt genoeg aan POJO's (Plain Old Java Objects), en daarmee is Spring veel beter testbaar en zijn de hoofdobjecten uit een Spring-applicatie heel veel beter te hergebruiken dan EJB-'objecten'. Met Spring kun je ook echt aan AOP (Aspect Oriented Programming) doen en dat is voor zowel Spring als AOP heel goed. Spring is OK! Spring lijkt inderdaad zo eenvoudig dat het wel eens heel succesvol zou kunnen worden. Maar, ook Spring is aan het groeien. Ook Spring wordt weer groter en groter. Mensen worden enthousiast van Spring en willen er ook heel specifieke problemen mee op kunnen lossen. Het is niet te houden.

Wordt J2EE al legacy? Het zal nog wel een tijdje duren. En tegen de tijd dat iedereen liever Spring dan J2EE gebruikt, zal er wel een geniaal alternatief voor Spring bedacht worden. Dat moet kunnen, Spring is mij nog steeds niet eenvoudig genoeg. Er zit nog steeds te veel XML en ingewikkelde tooling in.

*Daan Kalmeijer is docent consultant bij CIBIT-adviseurs | opleiders (e-mail: daan@cibit.nl).*