



Zonde van een tijdvak

Alsof David Beckham in de plaatselijke kerk plotseling Rebecca Loos tegen het lijf liep. Zo betrapte keek Lawrence Wilkes toen hij op een seminar werd aangekondigd als lid van het *Component Based Development & Integration* forum. 'Eigenlijk staat dat *CBDI* al een tijdje nergens voor', probeerde hij stamelend, 'het zijn maar vier letters'. Overtuigend klonk het niet.

Blijkbaar komt Wilkes er in het tijdperk van webservices niet graag voor uit dat hij nog maar kortgeleden één van de voornaamste pleitbezorgers was van component-based ontwikkeling. Begrijpelijk. Sommige zaken verzwijg je nu eenmaal liever. Zo heb ik ooit ergens in Brabant eens noodgedwongen tien seconden meegedaan aan een polonaise. Iets waar ik hoogstens in de meest intieme familiekring kort over wil praten (als volwassen man blij je dan hartverscheurend te kunnen huilen). Ook schreef ik in een ver verleden een boek over objectoriëntatie en C++. Dat heeft mijn carrière nog behoorlijk in de weg gestaan.

Of krijgt elk tijdvak de concepten die het verdient?

Vroeger was software engineering het domein van wetkundigen en elektrotechnici. Een beetje het bijvak van verstrooide bètagelerden. Systemen werden op strenge wijze uitgedrukt in termen van functies, parameters en variabelen. De Boleaanse Algebra vormde een rotsvast bastion dat programmeren tot een duister, semi-alchemistisch ambacht maakte. Als gebruiker hield je bij voorkeur grote afstand van de enorme complexiteit die blijkbaar nodig was om iets van een computer gedaan te krijgen.

Later werd ontdekt dat het ook wel handig was om gegevens op te slaan. Relationele databases deden hun intrede en SQL bracht de concepten waarmee zo handig mogelijk omvangrijke gegevensstructuren te lijf konden worden gegaan. Naast de klassieke programmeur ontstond een nieuw diertype in de vorm van de databasespecialist. Die hield er zijn eigen jargon op na en kon op besloten conferenties met vakbroeders gezellig discussiëren over de geneugten van viervoudig geneste joins. Niemand maakte daar een probleem van, zolang gewone mensen maar niet werden lastiggevallen.

Halverwege de jaren '80 deden PC's hun intrede en begonnen gebruikers snel benul te krijgen van waar computers allemaal voor kunnen worden gebruikt. Er werd geschreeuwd om inspraak en inzicht in het proces van systeemontwikkeling. De beide kampen in IT-land sloegen de handen zuchtend ineen en produceerden 4GL's om hun klanten sneller te kunnen voorzien van bruikbare applicaties.

Er waren er maar weinig die doorhadden dat zich onder tussen een dissidente onderstroom aan het vormen was. In de Palo Alto laboratoria van Xerox vond Alan Kay een programmeeromgeving uit (*'Smalltalk'*) waarmee kinderen gekleurde paarden over het beeldscherm konden laten dertelen. In plaats van het daarbij te laten werd Smalltalk de basis voor het verschijnsel *objectoriëntatie*. Ook in de nieuwe eeuw nog het onderwerp van een ware cultus waarbij het chic werd gevonden zoveel mogelijk objecten uit de werkelijkheid te modelleren en zo weinig mogelijk zaken die deden denken aan een werkend systeem of een dialoog. Combineer dat met toegankelijke concepten als *multiple inheritance* en *dynamisch polymorfisme*, en je had alle ingrediënten om systeemontwikkeling weer tot een echt, ondoordringelijk vak te maken.

Gebruikers bogen deemoedig het hoofd voor zoveel onpeilbare diepte. Daar hielp de komst van *componenten*, als een soort van versimpelde en platgeslagen versie van objecten, niets bij.

Pas met de introductie van XML en webservices lijkt een tijdvak aan te breken waarin het perspectief van de klant werkelijk dichterbij wordt gebracht. *Servicegeoriënteerde* systemen bestaan uit elementaire services die via het Internet op een gestandaardiseerde manier worden beschikbaar gesteld en geconsumeerd. Een simpele, overzichtelijke wereld waarin nieuwe, eenvoudige programmeertalen als *Water* hun intrede zullen gaan doen. Als organisatie denk je in termen van services die nodig zijn en services die aan anderen moeten worden aangeboden. Uitstapjes buiten de eigen bedrijfsmuren zullen daarbij steeds gebruikelijker worden. Wat ons weer terugbrengt bij de *Business-to-Business* escapades van David Beckham. Die kan in ieder geval nog altijd IT-consultant worden.