

VELDWIJK

Pluralitas non est ponenda sine necessitate

"Moet je je toch eens voorstellen hoeveel MIPS er in PC's zitten en nauwelijks worden gebruikt!" Dat was midden jaren tachtig de verzuchting van een mainframe-specialist bij RAET. Ik werkte toen bij dit door velen als oubollig beschouwde bedrijf, waar het voor managers gewoon was om naast de foto met gezin en hond een radiobuis of een tabelleermachine op het bureau te hebben staan. Ik vond deze collega vermakelijk. De tijd van de dure Mega Instructie Per Seconde was al voorbij en de wet van Moore bood uitzicht op een persoonlijke GIPS op ieders bureau. Wat ik, terugkijkend, toch van de RAET-cultuur heb meegekregen en waarvoor ik dankbaar ben, is gevoel voor het nut van monolithische oplossingen. Het woord 'Monolithisch' klinkt inderdaad oubollig en achterhaald, maar ik prefereer het boven het moderne alternatief, dat ik 'Verknipt' zou willen noemen. Terugkijkend op 20 jaar werken in en 10 jaar schrijven over ICT, ben ik mijn ex-collega's dankbaar voor wat ze me hebben meegegeven en waarover ik – vooral in DB/M – heb geschreven, zoals:

- "Geen aparte dictionary tools maar een dictionary als hart van je systeem – als je een flexibel systeem wilt";
- "Geen (fat-)Client/Server! – als je je nu (jaren negentig) vermogens aan PC clients wilt besparen en je wilt voorkomen dat investeringen weer over moeten worden gedaan";
- "Geen *Data Warehouse* of BI-oplossing voordat is vastgesteld dat een geïntegreerde oplossing niet haalbaar is";
- "Geen projecten waarin tegelijk ambitieuze generieke componenten en specifieke functionaliteit worden gebouwd – omdat zulke projecten in 100 procent van de gevallen falen".

Helaas heeft mijn geschrijf weinig invloed gehad – in elk geval niet op enige schaal. De verknipte oplossing voor elk automatiseringsprobleem is de norm en de kosten hiervan zijn fenomenaal. Vooral verontrustend is dat normale automatiseringsproblemen vaak te lijf worden gegaan met wat ik alleen maar een enorme conceptuele *overkill* kan noemen. Zo mocht een collega van mij enige tijd terug een mislukt project auditen. Het ging daarbij om een systeem dat niet veel anders doet dan maandelijks aangeleverde bestanden van verschillende organisaties met gelijksoortige sleutels controleren op overlap. Een banaler automatiseringsprobleem kom je niet vaak tegen: een systeem dat in essentie bestaat uit een *merge*-applicatie en dat vraagt om een handvol simpele batch – bestand-in/bestand-uit – programma's. Zoiets deden we eens met een RDBMS en een willekeurig front-end product.

Met zoiets simpels kan een moderne applicatie-architect echter

niet leven en de uitkomst was dus een systeem gebaseerd op Java, DB2, Websphere en MQ Series. Geen stomme batches voor de verwerking van oude gegevens maar *real time message queue management*. Dat is toch wel het minste wat je in de strijd moet werpen. Het gevolg: een kostbaar en niet onderhoudbaar systeem dat niet vooruit is te branden. Omdat probleemanalyse en software-aanpassing meer zouden kosten dan herbouw, adviseerde mijn collega om het systeem opnieuw te bouwen met behulp van simpeler concepten en minder hulpmiddelen. Dat advies werd overgenomen, maar gelukkig voor de architecten moest er bij het nieuwe systeem nog wat extra functionaliteit komen. Recent hoorde ik dat ook dit nieuwe project alweer in de problemen is gekomen. Veertig jaar na Cobol zijn sommige organisaties niet meer in staat om eenvoudige automatiseringsproblemen op te lossen. Gelukkig kun je elk eenvoudig probleem met goede concepten en tools voldoende moeilijk maken.

Ik denk en hoop dat veel lezers van deze column hun eigen organisatie niet herkennen in het extreme voorbeeld dat ik hier geef. Niet iedere organisatie is zo volledig de weg kwijt, maar ik geloof wel dat het verknipte denken en de conceptuele overkill bij het aanpakken van op zichzelf eenvoudige automatiseringsproblemen een breed voorkomend probleem is. Het is de tragedie van bladen als DB/M die willen schrijven over nieuwe technologieën en bedrijven als het mijne die betere systemen willen bouwen, dat het probleem vaak bij de oplossing wordt gezocht in plaats van omgekeerd en dat het de inhoudelijk betrokken technici vaak ontbreekt aan wat ik in mijn vorige column met de term *Judgment* heb aangeduid. Ik zou niet willen beweren dat het beter is om terug te gaan naar Cobol en het abonnement op DB/M op te zeggen, maar ik denk wel dat we dringend behoefte hebben aan criteria die ons beschermen tegen oplossingen die de maat van de problemen te boven gaan.

Gelukkig denken filosofen al eeuwen over dit probleem na. Het is tijd dat wij de door hun gevonden oplossingen in ons denken en werken gaan toepassen. Neem om te beginnen eens een half uurtje de tijd om iets op te Googelen over de filosoof wiens motto de titel van deze column vormt. Anders loopt het op een dag met ons net zo af als met RAET en PinkRocade.

René Veldwijk

Dr. R.J. Veldwijk (rene.veldwijk@faapartners.com) is partner bij FAA Partners, een onderdeel van de Ockham Groep.