



Ron Tolido

Bioritme

Als student informatica trof ik in een ver verleden ooit een docent programmeren met een onwrikbare levensvisie. “Wie in zijn programma een ‘Go To’ statement durft te gebruiken, timmer ik op zijn gezicht” vertelde hij ons in de eerste les. Het ontging niemand dat hij hierbij niet eens flauw glimlachte. Pas later kwam ik erachter dat ik met een adept van Edsger Dijkstra te maken had, de Nederlandse hoogleraar die vooral in Amerika triomfen vierde met zijn sterk wetenschappelijke benadering van het gestructureerd programmeren. Dijkstra schreef al in 1968 een artikel waarin hij betoogde dat het gebruik van de Go To tot slecht gestructureerde, moeilijk te onderhouden code leidt. Zijn ‘Go To Statement Considered Harmful’ werd een geveugeld begrip dat in lengte van jaren werd aangehaald, verbasterd en uiteindelijk geparodieerd.

Nu is er een einde aan een tijdperk gekomen. Want Dijkstra stierf onlangs, nog maar 72 jaar oud. En wie om zich heen kijkt, merkt nog maar weinig van zijn nalatenschap. De huidige generatie systeemontwikkelaars bedient zich weliswaar slechts zeer spaarzaam van de Go To (zelfs als ze met Visual Basic werken), maar verder worden de meest elementaire regels van gestructureerd en planmatig ontwikkelen met stampende voeten getreden. Geflankeerd door de kamermeisjes Haast en Spoed worden toepassingen bij elkaar geharkt, waarbij de gebruikersinterface, de bedrijfslogica en de databasetoegang één kluwen vormen. In het hijgerige tijdperk van de *One-Year CEO* is het blijkbaar hip om zoveel mogelijk bezig te zijn met gelikte flash-animaties en zo weinig mogelijk met programmaontwerp.

Dijkstra was nu eenmaal een vakidoot. Straf in de leer, alleen geïnteresseerd in pure oplossingen. Informatica zag hij als een wetenschappelijke uitdaging, niet als een middel om bedrijven meer geld te laten verdienen. Hij zal daarom ongetwijfeld niet hebben kunnen lachen om de losbolligheid van lichtgewicht ontwikkelmethoden als DSDM. Dat zijn de iconen van een nieuwe wereld waarin het onvoorspelbaarheid troef is en er steeds sneller resultaten moeten worden geboekt. Een wereld waarin de grenzen van een timebox veel meer bepalend zijn dan die van de functionele

specificatie. Een wereld waarin Steve Ballmer trots aankondigt dat zijn software voortaan automatisch een foutrapport verstuurt naar Redmond bij het optreden van een systeemcrash (‘we leren daar onzettend veel van’). Een wereld waarin we chique termen als *Refactoring en Retrospectief Documenteren* hebben uitgevonden om te legitimeren dat we soms eerst doen en dan pas denken.

Make no Mistake: een andere weg is er niet. Bedrijven die zich opportunistisch langs de economische pieken en dalen willen werken zullen een bij uitstek *adaptieve* houding moeten ontwikkelen: ze zullen zich als kwikzilver moeten kunnen gedragen op het moment dat kansen of verstoringen zich voordoen. En dat allemaal zonder dikke plannen vooraf. Bij zo’n houding hoort een IT-functie die even snel meebeweegt en dus per definitie vertrouwt op iteratieve processen en flexibele architecturen.

Het is dan wel zaak de bioritmen op elkaar afgestemd te hebben. Ik heb al te vaak IT-afdelingen gezien waarin de innovaties zich tot aan het plafond opstapelden (webservices, Extreme Programming), terwijl de bedrijfsvoering net bezig was de geneugten van koud en stromend water te ontdekken. Het verschil is dan eenvoudigweg te groot. Als een bedrijf geen onderscheidend voordeel ziet in wendbaarheid, dan heeft het weinig zin om de schuivende panelen van DSDM te introduceren. De eerste de beste workshop zal ont-aarden in door elkaar heen schreeuwende deelnemers die allemaal het volle pond aan functionaliteit eisen en volledige zekerheid vooraf. Net als vroeger eigenlijk. Stof dan vooral dat oude SDM-boek nog eens af en bestel wat extra pakken papier. Een beetje kastruimte voor alle specificaties kan ook geen kwaad.

Pas als starheid pijn gaat doen, is het tijd voor *agile* systeemontwikkeling. Tot dat moment kunnen we beter in detail plannen en ontwerpen. Dijkstra zou er trots op zijn, in het Grote Rekencentrum daarboven.