



Onder de Vulkaan

Waarom spuiten mensen muren vol? Waarom is de halve wereld aan de blog? Waarom schrijven mensen columns? Misschien vanuit een irrationele hoop op onsterfelijkheid; zoets als bij Pompeii waar anoniemi door hun graffiti nog steeds voortleven. Voor de oude Romein wiens gedachte niet verder ging dan "*Suspirium puellarum Celadus thraex*", moet het toch geweldig zijn om tweeduizend jaar later te worden gelezen door ons. Hoezo *fifteen minutes of fame*?

Helemaal mooi is het als je ook nog iets tijdloos nuttigs hebt te melden. Wie over ICT schrijft is dan vrijwel kansloos. Pak een oude Computable of (sorry) DB/M op en verbaas je over de waan van de dag. Maar heel weinig uitingen blijken een decennium of een halve eeuw later echt vooruitziend of zelfs waardevast. Misschien Orwell 62 jaar geleden met zijn wereld vol sensoren waarmee de staat alle menselijke communicatie monitort. Die gaat gelijk krijgen. Of een auteur in mijn informatica *reader* die zich ruim 25 jaar geleden zorgen maakte over de bureaucratisering die het gevolg zou kunnen zijn van de 'vooruitgang' op ICT-gebied. Die heeft zijn gelijk al. Idem Doctor Fred Brooks, die in 1986 in een beroemd artikel getiteld '*No Silver Bullet*' (zie wikipedia) betoogde dat de productiviteitsverbetering bij software tegen fundamentele grenzen aanloopt en dus op een dag ophoudt. Dat artikel is een *must read* voor iedereen jonger dan 40, niet alleen omdat het uitlegt aan welke fundamentele grenzen software-ontwikkeling is gebonden, maar ook omdat het een aantal veelbelovende ontwikkelingen van midden jaren tachtig bespreekt. Brooks dacht dat de productiviteit van software-ontwikkeling nog met maximaal [sic] een factor 10 zou kunnen toenemen. Meer is niet mogelijk omdat verdere productiviteitsverhoging daarna stuk loopt op de fundamentele beperkingen (*essential difficulties*) die aan software-ontwikkeling zijn verbonden. (Welke dat zijn? Lees het artikel!) Die factor 10 aan productiviteitswinst na 1986 zou zijn te boeken door het overwinnen van de niet essentiële beperkingen (*accidental difficulties*) die aan software-ontwikkeling zijn verbonden. Brooks vestigt daarbij zijn hoop op destijds nieuwe ontwikkelingen als hogere programmeertalen, Object Oriëntatie (alweer heel oud), AI, prototyping, programmageneratoren, betere opleidingen en pakketsoftware.

Hoe anders is het gelopen. Zelfs iemand met de grootste roze bril zal niet beweren dat de systeemontwikkeling nu tien keer sneller gaat dan 24 jaar geleden. Integendeel, de productiviteit is *afgenomen*, zeker in vergelijking tot de jaren negentig toen, naar ik vermoed, de productiviteit op zijn top zat. Ik werd daar

laatst mee geconfronteerd toen wij (voor het eerst) meededen aan een aanbesteding van de Overheid. Het ging om een systeem waarvoor onze aanpak en tooling bijzonder geschikt zijn. We offereerden daarom de Fixed Price onderdelen van de aanbesteding voor iets minder dan vier uur per functiepunt en kregen vervolgens vragen die duidelijk maakten dat dit ver, héél ver, onder de tarieven van de andere tien aanbieders zat. De norm voor Java systeemontwikkeling blijkt tegenwoordig in de orde van grootte van 8 of meer uren per functiepunt te liggen. Tja, volgende keer zullen we beter opletten. Maar ondertussen vermoed ik dat een Fixed Price project in 1987, met zeg Oracle 4, Forms 2.3 en Pro*C, ook was aanbesteed voor die 8 uur per FP. En midden jaren negentig waren scherpe aanbiedingen zeker niet hoger uitgevallen dan 5 uur per FP. De conclusie is dus dat de productiviteit van systeemontwikkeling de laatste tien jaar significant is teruggelopen, zonder dat dit veel vragen heeft opgeroepen. Voeg daarbij de epidemie van mislukkende projecten en het beeld ontstaat dat de *effectieve* productiviteit in vergelijking met tien jaar geleden met een factor 2 à 3 is teruggelopen – en dat dan niet alleen bij de overheid. De overheid is vooral bijzonder doordat er zoveel geld wordt besteed; waar helemaal niets uitkomt.

Heeft iemand dit voorzien? Niet de beroemde meneer Fred Brooks, al biedt zijn artikel wel houvast bij het begrijpen van waarom het zo mis is gegaan. Ook niet René Veldwijk. Die heeft wel een hele artikelserie geschreven over loze hypes, maar heeft nooit voorspeld dat de productiviteit zou gaan teruglopen. Geloof het of niet, maar er is in Nederland in elk geval één persoon geweest die dit verschijnsel heeft zien aankomen. En laat dat nou de meest onwaarschijnlijke persoon zijn; iemand die zijn brood verdient met het verspreiden van nieuwe ICT-ontwikkelingen. Niemand minder (en niemand anders) dan Rick van der Lans heeft ergens in de jaren negentig de stelling ingenomen dat de moderne post-4GL OO-tooling zoals Smalltalk (!) en Java te moeilijk zou blijken om mee te werken. Nu, dik een decennium later is het moment gekomen om Rick te eren voor zijn vooruitziende blik. Geef mij een muur onderaan de Merapi plus een spuitbus en ik vereeuwig Rick voor de generaties die na ons komen. *O tempera, o mores!*

René Veldwijk

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