

# Veldwijk

## *Softwarevergrijzing*

Kent u de term “Software Veroudering” nog? Voor mij was het midden jaren '80 de eerste van vele ICT termen die op mijn lachspieren werkte. “Hoe kon software nou verouderen?” Hardware ja, maar software is toch echt voor de eeuwigheid. Afijn, zo dacht ik destijds. Nu niet meer. Deels omdat je de term haast niet meer tegenkomt (Google maar!), maar vooral omdat ik nu vind dat de term Software Veroudering gepast is en onze aandacht waard is. Om te beginnen moderniseren we de term:

*Softwarevergrijzing* past meer bij 2006.

In letterlijke zin had mijn jongere ik gelijk: software veroudert niet. Ik kan mijn eerste Pascal programma's en mijn Wordstar afstudeerscriptie niet meer van mijn Osborne zeulable computer uit 1983 halen, maar dat komt door verouderde hardware. De software leeft in principe tot in eeuwigheid als bewijs voor mijn middelmatigheid als programmeur en warrige schrijfstijl. Softwarevergrijzing ontstaat niet alleen door tijdverloop maar vaak ook verkeerd menselijk handelen. Softwarevergrijzing begint bij software onderhoud: geen systeem zo flexibel of er zijn veranderingen die nopen tot code aanpassingen. Soms staan die aanpassingen haaks op de architectuur van de software en is een grote ingreep noodzakelijk, zelfs bij een in functionele termen kleine verandering. Zoals bekend wordt er in dit soort gevallen vrijwel altijd om het probleem heengeprogrammeerd en is de software veroudering een feit. Elke wijziging daarna vermindert de kans op de noodzakelijke fundamentele ingreep en elke wijziging wordt moeizamer door te voeren. U kent de uitdrukking “puist op puist”: softwarevergrijzing voor ordi's. Ook bij ogenschijnlijk probleemloze wijzigingen aan software treedt softwareveroudering op. Een fout zit in een klein hoekje en fouten zijn bijna zeker wanneer er geen goede documentatie is. Die constatering is ook recursief: foutloos programmeren zonder documentatie bij te werken is uitgesteld fout programmeren.

U denkt terecht dat de vorige 293 woorden een poging waren om een open deur in te trappen. Goed, maar beantwoord dan de vraag waarom we vrijwel niets doen aan softwarevergrijzing. Ik kom in elk geval maar zelden een systeem tegen dat meer dan een paar jaar oud is en waarvan de documentatie (nog) deugt. Het aardige is daarbij dat software veroudering met enige inspanning redelijk meetbaar is. Het enige dat nodig is, is het bijhouden van een functiepunten- en een kostentelling. Daarbij komen wat voor de hand liggende zaken zoals het mee laten tellen in de functiepunten van verwijderde functionaliteit. Niemand doet dit voor zover ik weet, maar neem gerust aan dat een systeem dat initieel wordt gerealiseerd voor, zeg, 6 uur per functiepunt na een jaar of tien vele tientallen uren per functiepunt heeft gekost. En ergens in die tijdspanne ligt dan het punt waarop herbouwen goedkoper is dan doormodderen. Want doormodderen en meer in het algemeen slechte besluitvorming zijn de norm. Niet ICTers in het algemeen en managers in het bijzonder houden heel lang vast aan het idee dat de waarde van een systeem oploopt met de hoeveelheid geld die er in wordt geïnvesteerd. Maar voorbij het punt waarop de softwarevergrijzing toeslaat en de investeringen per functiepunt de pan uit reizen neemt de waarde van een systeem juist snel af. Wie verstand van zaken heeft weet dit en doet er zijn voordeel mee. Ik ken een geval waarin een ICT bedrijf de eerste

versie van een door haar voor een klant gebouwd pakket terugkocht. Die versie had een fractie gekost van het bedrag dat er een paar jaar later aan was uitgegeven. ICT bedrijf en opdrachtgever dachten allebei dat ze een mooie deal hadden gesloten. De opdrachtgever kreeg de initiële bouwkosten terug. Maar het ICT bedrijf had berekend dat men beter aan de eerste versie kon doorbouwen dan aan het actuele systeem dat ondertussen vol zat met allerlei 'verbeteringen'. De moraal is dat het meten van softwarevergrijzing voor elke ICT afdeling een kernactiviteit zou moeten zijn – wat het dus niet is. De oorzaak moet denk ik ook worden gezocht in sociologie verkeerde incentives., maar zoals u weet beperkt deze columnist zich echter tot de harde kant van de ICT

Stel dat na publicatie van deze column iedereen opeens wakker wordt en we serieus gaan tellen, meten en wegen. Wordt de wereld dan beter? Het antwoord is “ja” en “JA!”. Functiepunten tellen is een weinig exacte aangelegenheid en functiepunten tellen gedurende het leven van een systeem is dat nog veel minder, maar zelfs al weten we maar heel ongeveer wanneer een systeem rijp is voor het verzorgingstehuis, dan al worden investeringsbeslissingen stukken beter. “Meten Is Weten” zetten sommigen en dat is mijn eerste, kleine, “ja”. “Meten Is Sturen” zeg ik en daarmee ben ik bij mijn grote “JA!” Meet de vergrijzing en systemen worden opeens in goede gezondheid oud. Softwarevergrijzing is niet zoals reguliere vergrijzing een natuurverschijnsel maar een keuze gebaseerd op niet-willen-weten. Als wij ICTers andere keuzes maken dan kunnen we allemaal lachend op ons 65e met pensioen.

René Veldwijk, [Rene.Veldwijk@FAAPartners.com](mailto:Rene.Veldwijk@FAAPartners.com), is partner bij FAA Partners.