

Ik was van plan om deze column te schrijven over het feit dat vele ontwikkelaars geen clou hebben van de organisatie waar ze voor ontwikkelen, dat programmeurs zich vaak alleen maar bezig houden met hun hobby's. Programmeurs die er een persoonlijke strijd van hebben gemaakt om één bepaalde programmeerconstructie bij alle andere programmeurs de strot in te duwen.

Waardevrij programmeren (bestaat niet)

Architecten die het introduceren van een middlewareplatform zien als het hogere doel in hun leven. Programmeurs en architecten die iets willen bereiken zonder dat daar ook maar enig bedrijfsbelang mee gediend is. Hobbyisme dus. Van elke regel code zou je je af moeten vragen of die het bedrijfsdoel dient. Zo niet – dan is het hobbyisme. Hobbyisten in overvloed.

Maar... toen werd ik gewezen op een veel ernstiger probleem. Tijdens een presentatie van een filosoof / ethicus (erg verfrissend om ook eens iemand anders te horen praten over ons vakgebied), werden de aanwezige IT-ers gewezen op het gegeven dat technologie nooit waarde vrij is. Hij liet ons zien dat elke technologische beslissing – elke introductie van een technologie – wordt genomen vanuit een bepaalde waardecontext, een culturele, sociale, religieuze of politieke context. Bovendien werd er betoogd dat organisaties en maatschappijen wezenlijk veranderen zodra daar nieuwe technologie geïntroduceerd wordt – soms subtiel, soms ingrijpend. Die twee gegevens gecombineerd leveren op dat wij – ontwikkelaars – geconfronteerd worden met een grote verantwoordelijkheid. Wat is jouw context? Wat is de context van jouw opdrachtgever? Wat zijn de consequenties van jouw product? Wat is de waarde die verstopt zit in jouw product? Wat doet jouw product met de organisatie waarvoor hij bedoeld is? Is jouw product wel een verantwoord product?

Laat ik eerst maar eens een voorbeeld geven van de waarde die een systeem kan bevatten. Afgelopen jaar was 'De intensieve menshouderij' (lezen!) een van de grote hits onder de managementboeken. Daarin werd ons voorge-

schoteld hoe onze organisaties processen centraal zetten en de kwaliteit van die processen boven de werkelijke kwaliteiten van de organisatie stellen. Hoe onze organisaties de kwaliteiten en het verantwoordelijkheidsgevoel van hun medewerkers negeren, die medewerkers alleen maar inzetten als domme radertjes in een geoliede machine. Radertjes die liefst geoutsourced zouden moeten worden. Een nogal somber beeld van organisaties die zich kapot staren op KPI's. Het boek gaat bijna nergens over ICT. Toch zou je kunnen (moeten?) stellen dat dergelijke organisaties vooral mogelijk gemaakt worden dankzij ICT - dankzij de ICT die wij bouwen! Beseffen wij ons wel dat wij helpen een dergelijke organisatie in te richten? Zijn onze opdrachtgevers zich daarvan bewust? Is het onze verantwoordelijkheid om de opdrachtgevers te wijzen op de invloed die onze systemen gaan hebben? Moeten wij niet met alternatieven komen?

Na de presentatie van de filosoof bleek vrijwel iedereen het eens met zijn stellingen. Een aanzienlijk deel van de aanwezigen echter was expliciet niet van plan de consequenties van die stellingen te accepteren als een persoonlijke verantwoordelijkheid. Zij zien zichzelf als iemand die niets anders doet dan requirements van opdrachtgevers uitwerken in systemen. De verantwoordelijkheid ligt volgens hen volledig bij de opdrachtgever. Daarmee gaan ze er aan voorbij dat zij zelf ook een culturele, politieke en religieuze context met zich meedragen en dat die context zich ook laat zien in de systemen die zij ontwikkelen. Dat is erger dan hobbyisme. Hobbyisme is gewoon nutteloos, maar niet verantwoordelijk willen zijn voor je eigen daden is misdadig.

Daan Kalmeijer

is docent consultant bij DNV-CIBIT
(e-mail: Daan.Kalmeijer@DNV.com).

