

Ik heb het de afgelopen tijd veel te druk gehad om naar de film te gaan en de komende weken ziet het er ook niet best uit. En dat terwijl er nou net één film is die ik zo graag zou willen zien: *The Hitchhikers Guide to the Galaxy*. Nu dat niet gaat lukken luister ik de laatste tijd veel naar het oorspronkelijke hoorspel. Waar vele films één wereldberoemde tekst kennen, kent *The Hitchhikers Guide* geen zin, geen woord, maar één getal: 42.

## Tweeënveertig

“Hasta la vista, baby”, “Go ahead, make my day”, “The force will be with you” en “Are you talking to me?” zijn zinnen die je direct aan de bijbehorende film doen denken. “42” is alles wat nodig is om de fans van *The Guide* de oren te laten spitsen. “42” gaat om het volgende: Er wordt een supercomputer gebouwd om uit te kunnen rekenen wat de zin van het leven is, “the Ultimate Question of Life, the Universe, and Everything!” De supercomputer verklaart dat hij het antwoord wel kan berekenen, maar dat hij daar 7.500.000 jaar over zal moeten doen. Na 7.500.000 jaar komt hij met een antwoord (“but you won’t like it ...”). Tot ieders verbazing is het antwoord: 42! De supercomputer vertelt dat het nu tijd is om, door een nog grotere computer, uit te laten rekenen wat er bedoeld was met de ultieme vraag. Dit is een situatie waar wij software-ontwikkelaars dagelijks mee te maken hebben. “Hoe lang denk je dat jullie nodig hebben om ons nieuwe systeem te bouwen? 16 maanden? Prima, dan schrijf ik op dat het systeem klaar is op 27 december 2007”. Er wordt 16 maanden geploeterd om een deadline te halen, niet om een systeem te bouwen. Het

‘systeem’ dat opgeleverd wordt is niet veel meer dan een vage verwijzing naar de oorspronkelijke vraag. Pas na het bouwen gaan we nadenken over de oorspronkelijke vraag.

Ik ga er van uit dat software bouwen onvoorspelbaar is. Volledig onvoorspelbaar! De bouw van innovatieve, ingrijpende, spannende, wereldverbeterende business-software is niet te voorspellen. Wel hebben we ondertussen zo veel platte, administratieve formuliervervangende COBOL-achtige programma’s gemaakt dat we de bouw daarvan aardig kunnen voorspellen. Maar ja, dat is dan ook niet veel meer dan letterlijk papieren businessprocessen vervangen door een softwarevariant daarvan. Software kan, en moet, heel veel meer betekenen. Als software de business ingrijpend verandert, dan kan niemand op de millimeter voorschrijven hoe die software eruit moet zien. Softwarebouwers en business mensen zullen samen moeten ontdekken hoe de software moet werken.

De meeste softwaretheoretici denken hier heel anders over. Zij denken dat wij gewoon niet goed genoeg voorspellen. Ze zijn op zoek

naar de definitieve formule voor het plannen van software. Om voorspelbaar te kunnen plannen, moet het proces van software bouwen vastgelegd zijn. “Eerst doe je A, dan B en dan C. Waarbij A twee keer zo lang duurt als B”. CMM(-I) is gebaseerd op dit waanidee van de voorspelbare software. Daarom legt het zo’n nadruk op het ‘volwassen’ maken van het ontwikkelproces. Het gaat er van uit dat, wanneer je weet wat je als eindproduct wilt hebben, je terug kunt rekenen naar alle stappen die daar aan vooraf horen te gaan. Maar dat is natuurlijk onzin. We doen juist al die stappen omdat we nog niet weten wat de definitieve software moet doen. We kunnen vooraf dus niet zeggen hoe lang die processtappen gaan duren. CMMI verbetert maar één soort software: platte, oninteressante, formuliervervangende, administratieve software. Voor alle innovatieve software is CMMI (en andere vormen van procesfetisjisme) de dood in de pot.

*Daan Kalmeijer is docent consultant bij  
CIBIT adviseurs | opleiders  
(e-mail: daan@cibit.nl).*