

VAN DER LANS

Wat drijft de IT-specialist?



Hier en daar lopen IT-projecten nog wel eens fout. Het ene project blijkt wat duurder te zijn dan gepland, een ander duurt veel langer dan verwacht en een derde levert geheel niets op. Vreemd, want in de afgelopen dertig jaar zijn heel wat nieuwe technologieën en technieken zoals SOA, OO, MDM, NoSQL, Cloud en CBD geïntroduceerd die allemaal beloofden dat alle problemen als sneeuw voor de zon zouden verdwijnen.

Zijn falende projecten moeilijk te vinden? Niet echt. Tik maar eens bij Google in 'mislukt IT project'. Je vindt dan al snel een forse lijst, zoals het ministerie van VROM dat eind 2007 het automatiseringsproject VIDI staakte. De teller stond toen al op 16,6 miljoen euro. In juni 2008 blies het UWV een grootschalig ICT-project af ter waarde van 80 miljoen euro. In 2008 en 2009 schreef Bankier Van Lanschot 55 miljoen af op een mislukte modernisatie van de back-office systemen. En in 2010 stopt het Belgisch ministerie van Justitie met de ontwikkeling van het nieuwe gevangenissysteem Cajis, waaraan al 12 miljoen euro was uitgegeven. Als u grotere bedragen wilt zien, ga dan naar spectrum.ieee.org/computing/software/why-software-fails.

Maar waarom lopen IT-projecten soms zo glansrijk fout? Wie is schuldig? Zijn het de gebruikers, onze producten, slecht gespecificeerde gebruikerswensen, veranderende wensen, of managementtechnieken? In het boek *Crash, ten easy ways to avoid a computer disaster* geeft Tony Collins elf redenen waarom projecten mislukken.

Eigenaardig genoeg wordt zelden gewezen naar ons, de specialisten. Wij doen dat in ieder geval niet. Zou het niet nuttig zijn als wij onszelf eens aandachtig in de spiegel bestuderen? Mijn mening is dat als we kritisch naar onszelf kijken, we kunnen concluderen dat een aantal principes ons leidt. Die principes zouden ook wel eens de reden kunnen zijn van het falen van een project.

Specify-once. Er is een sterke neiging om systemen te ontwikkelen waarbij elke specificatie maar één keer wordt ingevoerd (en uiteraard meerdere keren gebruikt). Het grote voordeel hiervan is dat als de specificatie aangepast moet worden, dat maar op één plek hoeft te gebeuren. Maar is er wel eens uitgerekend wat de kosten en baten hiervan zijn? Krijgen we hierdoor niet teveel afhankelijkheden en een inflexibel systeem? Het hele idee is trouwens een idee-fixe, want de organisatie hoeft maar te fuseren en we hebben weer overal duplicaten.

Tool-addiction. Veel specialisten hebben hun favoriete ontwikkeltaalen, producten, methoden. Veel zijn er zelfs aan verslaafd. Zij zullen altijd proberen welk probleem dan ook met hun favoriete speeltje op te lossen. Ik weet zeker dat dit ook tot mislukkingen kan leiden, puur omdat het verkeerde product is ingezet.

Free-of-charge. Vaak weten wij niet wat de kosten van een technische oplossing zijn. Vraag maar eens aan honderd BI-specialisten wat een datamart ongeveer kost. De meesten hebben geen idee, zelfs de bouwers niet. Toch nemen we dagelijks architecturale en ontwerpbeslissingen die enorme financiële consequenties hebben. Maar daar kijken we amper naar. Alsof er aan een oplossing geen prijskaartje hangt.

Performance-dictates. Veel argumenten om een bepaalde oplossing te verdedigen worden vaak van de tafel geveegd als het gevoel bestaat dat het ten koste van de performance gaat. Performance is heilig. Alles moet hiervoor wijken. Maar weet de organisatie wat de prijs is die zij moet betalen om die snellere oplossing te krijgen? Specialisten kiezen altijd voor de snelste oplossing, maar maken het daardoor soms onnodig duur. Zouden ze dat thuis ook doen?

Hype-sensitivity. Het is u waarschijnlijk al opgevallen: we willen vaak direct de nieuwste technologie inzetten. Maar dat brengt risico's met zich mee. Hoelang iets duurt en hoeveel het gaat kosten, is meestal niet goed duidelijk, want eerst moet er ervaring opgebouwd worden.

Wat ik met deze column niet wil zeggen, is dat IT-specialisten altijd de schuldigen zijn, maar dat we eerlijk moeten zijn en moeten toegeven dat wij wel ons steentje bijdragen. Laten we eens beginnen ons eerst wat bewuster te worden van de bovengenoemde principes. Misschien moeten we aan sommige daarvan iets minder waarde schenken. Het hoeft niet altijd met een nieuw product, het hoeft niet altijd de snelste oplossing te zijn, en het hoeft niet altijd de mooiste oplossing zijn. Goed is vaak goed genoeg.

Rick van der Lans is zelfstandig IT-consultant.