



**Ron Tolido**  
tolido.blogspot.com

**Als het gaat over klusvaardigheden, ben ik zonder meer het lachertje van de familie. Ik heb weliswaar twee rechterhanden, maar daar staat tegenover dat ik linkshandig ben. Mijn broer bouwt achteloos fluitend complete keukens op waarbij hij ook nog eens louter voor de sport het loodgieterwerk regelt en de elektrische bedrading vernieuwt. Zelf begin ik al paniekerig om me heen te kijken als er een Billy boekenkast in pakketvorm naar binnen wordt gedragen.**

## Bob de Sloper

**A**n de andere kant heb ik een onvermoed talent voor slopen. Geef mij een voorhamer en zelfs de meest volgestouwde uitdragerij ziet er binnen een paar uur uit als een schoongeveegde meditatiekamer. Een onderschatte maar wel degelijk nuttige vaardigheid die – als ik me niet vergis – ook in het IT-landschap een steeds grotere rol gaat spelen.

Want zeg nou zelf: we leiden onze systeemontwikkelaars nog steeds vooral op als bouwers. Dat doen we vanuit het bijna schattig naïeve uitgangspunt dat software wordt ontwikkeld vanuit een sappige, groene weide. Bijna al onze vernieuwingen in het vakgebied zijn erop gericht om nog beter, sneller en slimmer te bouwen. Vanuit het niets. Daartoe proppen we onze gereedschapskist vol met versnellende hulpmiddelen als Model Driven Architecture, frameworks, componenten, services en Domain Specific Languages.

En dan maar oprecht teleurgesteld zijn als die gereedschapskist steeds minder vaak opengaat.

De reden? Alle rommel die de bouwers voor ons - of misschien wel wijzelf - hebben achtergelaten. Het gros van het budget van de gemiddelde IT-afdeling gaat ondertussen op aan het in de lucht houden van verouderde systemen. Die zijn soms geschreven in programmeertalen die alleen nog vanuit de overlevering bekend zijn (zo wordt het kernsysteem van een niet nader te benoemen Zweedse overheidsorganisatie alleen nog doorgrond door een 74-jarige programmeur; deze wordt met staalpillen, massages en dagelijkse bloedtransfusies zoveel als mogelijk in conditie gehouden).

En de boterberg aan onwrikbare en roestige software groeit nog steeds elk jaar door. En daarmee ook de frustratie dat er niet genoeg ruimte overblijft om nieuwe applicaties te bouwen. Je zal nog

zo'n mooie keuken in je hoofd hebben, als je tussen de ingestorte muren en het verzakte dak nog niet eens je ladder neer kunt zetten, kom je nergens met al die handigheid en fraaie intenties.

Mag ik daarom een warm pleidooi houden voor het terugbrengen van het goede, oude ambacht van slopen in de systeemontwikkeling? Alleen al met een goed timmermansoog kunnen kijken naar de bestaande portfolio van applicaties, dat zou een goede stap vooruit zijn. Welke delen van systemen leveren waarde aan de bedrijfsvoering en welke niet? Hoe verhoudt die waarde zich tot de hoeveelheid tijd die erin moet worden gestoken? En in welke mate is de onderliggende technologie nog relevant en uitbreidbaar?

Dat alles geeft inspiratie voor het in elkaar zetten van de te volgen sloopstrategie. Alles weggooien, de data migreren en overgaan op een standaardpakket bijvoorbeeld. Of de meest wankele elementen isoleren en vervolgens stuk voor stuk herbouwen op een nieuw platform. Of de proceslogica en bedrijfsregels uit de code plukken en onderbrengen in aparte Business Process Management en Business Rule systemen. Of met behulp van geautomatiseerde tools de structuur van de bestaande software ontwarren en versimpelen. Of – als niet anders werkt – het oude systeem zoveel als mogelijk stabiliseren en daaromheen nieuwe interfaces en façades opbouwen (ook wel bekend als de Parijse Variant).

Het zijn zomaar wat voorbeelden van de sloopvaardigheden die je als systeemontwikkelaar tegenwoordig in zo'n beetje elk serieus project nodig hebt. We kunnen er wat mij betreft in het vakgebied niet genoeg aandacht aan besteden, zowel in het onderwijs als in onderzoek en innovatie.

Kunnen we het slopen? Ja, dat kunnen we! «