

# Hoofdkantoortoepassingen in .NET bij KBC

“OOK AGILE PROGRAMMING MOET JE AGILE TOEPASSEN.”

Twee ICT-werelden bij KBC met elk een eigen visie op en methodiek van programmeren. De mainframeomgeving met watervalmethode en agile programming voor de .NET-omgeving.

In België is KBC de op drie na grootste bankverzekeraar. De organisatie is een onafhankelijke, middelgrote bankverzekeraar voor particulieren en middelgrote ondernemingen, en is actief in het thuisland België en een aantal Europese landen waaronder Hongarije, Polen, Slowakije, Slovenië en Tsjechië. KBC telt ongeveer 50.000 werknemers en meer dan 12 miljoen cliënten. De ICT-afdeling telt in België ongeveer 2000 medewerkers, van wie er ongeveer 1200 werken voor ICT Ontwikkeling.

Geert Vandezande is diensthoofd OKA en verantwoordelijk voor hoofdkantoortoepassingen bij KBC. Geert: “OKA is een dienst binnen de ICT-directie ODK (Ontwikkeling Distributiekanaal). Wij zijn voornamelijk verantwoordelijk voor hoofdkantoortoepassingen, geen kritische systemen, maar typisch toepassingen om backoffice-processen te ondersteunen. Zo’n twee jaar geleden hebben we binnen KBC een grote oefening gedaan om een nieuwe softwarestraat op te zetten. Enerzijds was er reeds de traditionele high-end mainframe- en Java-omgeving voor de ontwikkeling van de core-systemen en anderzijds hadden we ook softwareontwikkeling bij de business-entiteiten zelf, en end-user computing cellen voor de niet-kritische applicaties. Er tussenin was er een Visual Basic-ontwikkelstraat ontstaan. Die was eigenlijk nooit officieel gepositioneerd. Ongeveer twee jaar geleden zijn we op zoek gegaan naar een alternatieve omgeving voor deze Visual Basic-straat waarmee we goedkoop, snel, efficiënt en flexibel toepassingen kunnen ontwikkelen. Die nieuwe softwarestraat moest er komen naast de robuuste mainframeomgeving met zijn procedurele watervalmethode, uitgebreide analyses en het strikte beleid met zo drie a vier keer per jaar een updatemogelijkheid. Voor de nieuwe softwarestraat wilden we juist een andere werkwijze die meer past bij onze doelstellingen. We hebben vrij snel de beslissing genomen dat we het volgens ‘agile programming’ gingen opzetten, als opvolger voor Visual Basic was NET een voor de hand liggende keuze.”

## .NET-softwarestraat

“De keuze voor Microsoft .NET is weloverwogen”, vervolgt Geert, “want je kunt ook agile programming met Java doen. Java wordt bij KBC gebruikt voor de bouw van de web-based presentatielaag van de kritische applicaties die zelf op het mainframe gehost worden. Een belangrijke overweging om voor .NET te kiezen was dat we een strikte scheiding hebben tussen de mainframe- en de op Unix gebaseerde omgeving met de kritische toepassingen en de kant met de minder kritische.”

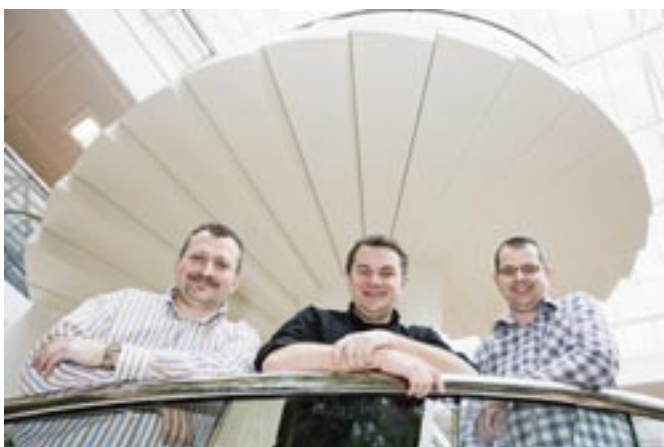
Geert: “We zijn in 2005 met het opzetten van een ontwikkelstraat begonnen. Als eerste hebben we een RFP uitgeschreven naar een aantal leveranciers en getracht er een te kopen. Met één van die leveranciers hebben we een pilootproject gedaan dat succesvol is verlopen. De nieuwe Visual Studio 2005 en Visual Studio Team Foundation Server kwamen op dat moment echter op de markt en de aange-

boden ontwikkelstraten waren hierop niet voorzien. We hebben er dan voor gekozen om zelf een ontwikkelstraat inclusief framework op te zetten. We hadden namelijk een concreet project klaar staan.” Met de hulp van Microsoft Consultancy Service België is onder hoge tijdsdruk de eerste versie van het framework opgezet. Met deze eerste versie is het project van start gegaan en in 2006 is het framework verder uitgebreid.

“Met de .NET-softwarefactory die we nu hebben kunnen we snel en efficiënt toepassingen ontwikkelen”, legt Jan Laureys uit. Hij is senior teamleader ICT bij KBC en verantwoordelijk voor het aansturen van de groep mensen die .NET-projecten doen. Jan: “Dat is compleet anders dan de softwarestraat aan de mainframekant. De keuze aan welke kant we gaan ontwikkelen is policy-driven. Op basis van de policy wordt naar de behoefte gekeken die er uit de businessunit komt. Is de te bouwen applicatie core of non/core, wat zijn de SLAs die verwacht worden?” Geert voegt hieraan toe: “We hebben bij KBC heel strikt de lijn getrokken tussen de twee omgevingen. Hoewel er interoperabiliteit mogelijk is tussen de .NET-omgeving en de mainframe-/Java-omgeving passeren we die grens niet.” Jan Laureys: “Wat mede onze keuze voor de Microsoft-tools heeft bepaald is dat we met een offshore partner werken: Value Source, een KBC-dochter. We zijn met Team System in staat om virtuele teams op te zetten. We werken samen aan een project met ontwikkelaars die hier 10.000 km vandaan zitten. Team System faciliteert deze teams. We kunnen na één jaar vaststellen dat het werkt!”

## Agile Programming

“Als je voor agility kiest moet je dat steeds blijven verdedigen”, vertelt Geert Vandezande. “Ik noem bijvoorbeeld testen. We hebben bij KBC een competence center dat zich puur en alleen met testtools bezighoudt. Dat competence center wil voor het hele bedrijf bepalen welke testtools ze moeten gebruiken. Dat is natuurlijk goed maar voor ons zijn dat waarschijnlijk niet de juiste tools.” Jan vult aan: “De manier van werken, om alles vooraf uitgebreid te documenteren is zeker niet de onze. Documenteren is goed, maar het moet wel in lijn zijn met de toepassing. Het mag geen documenteren worden in plaats van programmeren. Agile programming houdt voor mij ook in dat we constant kijken of we nog efficiënt te werk gaan. We onderzoeken regelmatig waar en wat we kunnen verbeteren. Zo zijn we met UML begonnen, maar het bleek dat de communicatie met eindgebruikers vaak lastig was. We zijn overgegaan van use-cases naar user-stories, die minder strikt en beter begrijpbaar zijn voor de businesspartijen. We tekenden nu ook schermen en minder UML-diagrammen. We zijn binnen Team System gestart met MSF maar zijn vrij snel op de Scrum-template van Conchango overgestapt”, zegt Jan. “Agility is de leidraad, maar we zijn ook weer niet fundamentalistisch. Ik bedoel daarmee dat we ook op dat vlak agile zijn. We passen alle aspecten



Afbeelding 1. Van links naar rechts: Jan Laureys, Geert Vandezande, Peter Bauwens.  
(Fotografie Hans Oostrum)

van agile programming toe, maar hebben wel ontwikkelaars offshore aan het werk. We krijgen vaak de vraag: hoe kun je nu agile programming toepassen (agile programming met weinig documentatie, peer programming, stand-up meetings en noem maar op) met ontwikkelaars die 10.000 kilometer verderop werken? Het antwoord is: Ja, dat is mogelijk. De technologie en tools staan het ons toe: iterative, code check-in, daily build, et cetera, we passen het allemaal toe. Als een ontwikkelaar in India zijn code incheckt, ziet de technical lead dat hier in België onmiddellijk. Hij kan bijvoorbeeld direct een code-review doen. Agile programming moet je ook agile toepassen.”

## Mensen

Geert: “We hebben ook de teams aangepakt. We zijn in 2005 met drie man begonnen en dat is uitgegroeid tot 34 medewerkers. We zijn bij KBC typisch volgens de lines of business georganiseerd. Mijn dienst staat daar haaks op, omdat wij voor alle businessunits werken. Wij doen ook kortlopende projecten. In de mainframewereld duurt een project snel 200 manweken, wij spreken over projecten van gemiddeld 50 manweken. In het afgelopen jaar hebben we 14 projecten gedaan. We zijn nu in staat om in ongeveer 10 à 20 manweken een applicatie te bouwen.”

“Nu zijn we bezig met de opsplitsing van teams naar een model, met welke je een service center hebt dat weer een aantal competence centers heeft. Eigenlijk net zoals de pattern & practice-teams bij Microsoft. Daarnaast hebben we delivery centers die verantwoordelijk zijn voor het bouwen van de software”, zegt Peter Bauwens. Hij is de Technical Lead van het Service Center .NET. Peter: “We hebben drie rollen gedefinieerd: projectleider, Scrum-master of lead developer en voor de rest developers. De traditionele analist valt er uit en ook de tester. Bij één van de eerste projecten was de analist nog bezig met het analyseren, terwijl de code al werd opgeleverd. Dat werkte niet. Via Team System communiceren de mensen. Alle requirements, bv. user-stories worden erin gestopt. Het mooie van Team System is dat als je taken gaat toekennen aan mensen, ongeacht waar ze zitten, alles weer teruggekoppeld wordt aan de requirements. Alles hangt aan elkaar, de code die ze inchecken, de bugs die ze fixen. En door goede rapporten heb je een volledig overzicht. De traditionele vragen van een projectleider over de status komen niet meer voor. Het grappige is wel dat vroeger alleen de projectleider geïnteresseerd was in de status, tegenwoordig is bijna iedere developer geïnteresseerd. De teamleden zijn vooral ook geïnteresseerd in de teamsnelheid.” Jan: “We merken wel dat de mensen in het core-team nog niet aan onze snelheid gewend zijn. Want bij veel projecten zijn er afhankelijkheden, zoals op het gebied van databases. Wij werken met maandelijkse releases, terwijl bij de core-teams de releasecyclus van de mainframe omgeving wordt aangehouden. We merken dat er lieden aan de core-kant met jaloezse ogen naar onze omgeving met Team System kijken. We zijn trouwens gezamenlijk aan het kijken welke onderdelen van ons aan de core-kant toegepast kunnen worden zonder bij hen het formele proces aan te passen.”