

# Voorwoord

## Vijf jaar .NET Framework

**B**ijna vijf jaar geleden kwam ik voor het eerst in aanraking met het .NET Framework. Ik werkte toen al een tijdje als consultant bij Microsoft en was na een lange geschiedenis met C, C++ en Java al behoorlijk vertrouwd met Visual Basic 6 als programmeertaal en COM als middle-ware-technologie. Waar je als Microsoft-developer niet zonder kon was het boek 'Effective COM' van Don Box. Het was van voor tot achter gevuld met de kleinste COM-details en essentiële tips. In die tijd hielden we regelmatig de - inmiddels beruchte - 'COM Cruises', waar cursisten afgezonderd in een vakantiepark vijf dagen lang werden ingewijd in alle ins en outs van COM. Aan het eind van die vijf dagen kregen de moegestreden cursisten dan het felbegeerde T-shirt 'COM is LOVE'. Als programmeur van een high-level programmeertaal als Visual Basic 6 was het toch noodzakelijk om behoorlijk wat kennis te hebben van de onderliggende omgeving om een fatsoenlijk presterende applicatie te schrijven.

Toen in 2000 meer bekend werd over de nieuwe ontwikkelomgeving werd al snel duidelijk dat het niet ging om een nieuwe naam voor de bestaande COM-technologie of een nieuwe release met wat extra features. Het ging om een compleet nieuw platform waarop goede zaken uit bestaande omgevingen en talen waren samengebracht, maar de bedenkers zijn niet bang geweest om nieuwe ideeën te introduceren.

Nieuwe termen en afkortingen vlogen ons telkens weer om de oren tijdens elke sessie op de PDC, TechEd en andere conferenties. Het viel de oplettende ontwikkelaar direct op dat we de beschikking kregen over een framework waarmee het mogelijk was snel en effectief een applicatie te kunnen bouwen. Er ontstonden meteen heftige discussies over wat technisch de beste programmeertaal zou zijn - VB.NET of C# - terwijl het concept van .NET dit nu juist minder belangrijk maakt. Grappig is dat deze discussies tot op de dag van vandaag nog steeds gevoerd worden. Kennis van het .NET Framework en de bijbehorende class-library is naar mijn mening nog steeds de belangrijkste factor voor het slagen van een .NET-project. Het juiste gebruik van onderdelen van het .NET Framework zou moeten resulteren in een applicatie die voor minstens voor 90% uit businesscode bestaat.

Met de komst van nieuwe technologieën als XML en webservices werden nieuwe zaken mogelijk die tot dan toe als technisch moeilijk haalbaar werden gezien. Helemaal ondenkbaar toentertijd was het delen van informatie en het aanroepen van functionaliteit op verschillende platformen. In het begin waren er nog heel veel discussies en bezwaren over de performance, overhead en beschikbaarheid van dit soort zaken, maar er zijn nu nog maar weinig nieuwe projecten waar ze niet worden toegepast.



Hetzelfde is ook van toepassing voor het hele .NET Framework. Er worden nog maar weinig nieuwe projecten gestart waar geen gebruik van het .NET Framework wordt gemaakt voor het bouwen van applicaties op het Windows-platform.

Het is waarschijnlijk niemand ontgaan dat Microsoft dit jaar Visual Studio 2005 en versie 2.0 van het .NET Framework gaat uitbrengen. Zoals ook bij de vorige releases het geval was zullen de gebruikers vroegtijdig toegang krijgen tot bètaversies van deze producten. De overweldigende set van tools zal direct opvallen. Hiermee kunnen nu ook architecten en testers hun applicaties ontwerpen, bouwen en testen. Developers kunnen al tijdens de bouw van de software de kwaliteit van hun code beoordelen en waar nodig verbeteren door gebruik te maken van code analyzers en profilers die in hun ontwikkelomgeving zijn geïntegreerd. Class designers maken het eenvoudig om classes toe te voegen of te wijzigen en snel door een objectmodel en onderliggende code te navigeren.

Minder zichtbaar, maar zeker niet minder belangrijk, zijn de Framework-classes en nieuwe features in ASP.NET. Dankzij feedback van gebruikers, en door analyse van de .NET-code die de afgelopen jaren werd geschreven, is het nieuwe .NET Framework aangepast en uitgebreid. Het percentage van 90% businesscode in een applicatie zal hiermee naar mijn mening nog hoger kunnen liggen.

ASP.NET is een onderdeel van het framework waarbij dit zeker het geval is. De hoeveelheid code die bijvoorbeeld nodig is voor het tonen van informatie uit een database of het valideren van gebruikers is gereduceerd tot een absoluut minimum. Had ik bij de eerste ASP.NET 2.0 sessies nog twijfels over de claim van 70% codereductie, na mijn eerste hands-on ervaringen met deze versie ben ik overtuigd dat dit zeker haalbaar is.

Het is duidelijk dat het krijgen van goede technische informatie belangrijk is voor het succesvol uitvoeren van .NET-projecten. Ik ben er van overtuigd dat wij jullie via sessies, evenementen en het .NET Magazine zo goed mogelijk informeren. Zodat jullie met de nieuwe versies van Visual Studio en het .NET Framework jullie projecten succesvol en nog efficiënter kunnen uitvoeren.

Veel leesplezier!

**Arie Leeuwesteijn**  
Developer and Platform Group  
ariele@microsoft.com