

# Ontwikkelen en design gaan nu beter samen

## WERKEN MET VOORBEELDDATA IN EXPRESSION BLEND 4

Willem Meints

Het ontwerpen van een user interface vergt steeds vaker meer dan alleen het ontwikkelen van een aantal schermen met daarop een aantal invoervelden. De gemiddelde gebruiker zit langer achter een computer en werkt ook langer met een bepaalde applicatie. Als je een goede applicatie wilt bouwen zul je er voor moeten zorgen dat een applicatie prettig is in het gebruik, maar dat het gebruik ook effectief en efficiënt kan gebeuren.

**Bij het ontwerpen** van deze gebruikerservaring is een samenwerkingsverband tussen een interaction designer en een ontwikkelaar erg belangrijk. De designer draagt hierbij zorg voor een goede interactie en grafisch ontwerp en de ontwikkelaar is verantwoordelijk voor het correct implementeren van het benodigde gedrag en het koppelen van de juiste data. Eerder was het zo dat de schermontwerpen eerst op papier of in een tekenprogramma werden uitgewerkt door de designer en daarna werden geïmplementeerd in een door de ontwikkelaar gekozen frontend technologie. Hoewel er al veel applicaties op deze manier succesvol zijn geïmplementeerd is deze manier van samenwerken niet altijd even effectief.

Microsoft heeft deze uitdaging onderkend en met Expression Blend een oplossing bedacht waarmee de designer de user interface kan ontwerpen, waaraan de ontwikkelaar in Visual Studio gedrag kan toevoegen. Het doel is om de samenwerking tussen de ontwikkelaar en de designer te bevorderen en te stroomlijnen met tools die ervoor zorgen dat het minder energie en tijd kost om een goede user interface te implementeren.

Verder wordt het Model View View Model (MVVM) design-pattern vaak toegepast voor het scheiden van presentatie en gedrag. Het MVVM design-pattern stelt je in staat om het ontwerp te scheiden van het gedrag door een zogenaamd viewmodel te introduceren. De ontwikkelaar implementeert het gedrag in dit viewmodel, terwijl de designer de view ontwerpt.

Deze nieuwe manier van werken klinkt mooi, maar brengt ook nieuwe uitdagingen met zich mee. Hoe zorg je er nu voor dat een designer een view kan maken waarbij hij wel de data kan gebruiken uit het viewmodel om tegenaan te ontwerpen?

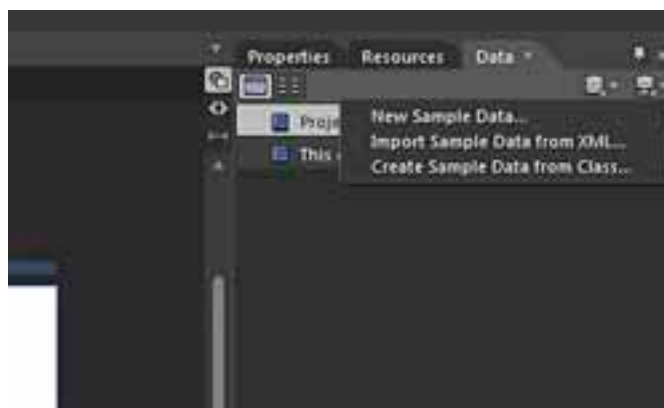
In dit artikel laat ik zien hoe je met deze uitdaging om kan gaan door gebruik te maken van de mogelijkheid om voorbeelddata te definiëren in de Expression Blend tools.

### Ontwerpen zonder back-end

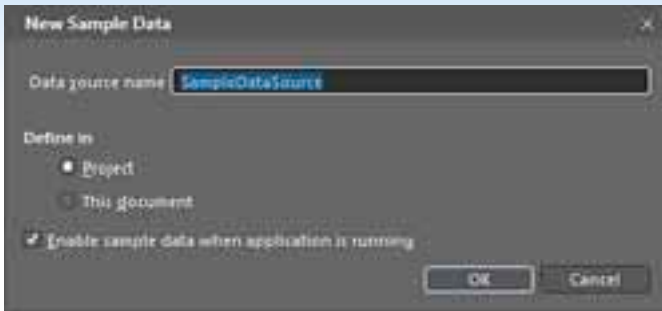
Niet elke user interface begint met het implementeren van het model waar later de schermen aan worden gekoppeld. Het komt vaker voor dat het ontwerpproces andersom wordt ingestoken. De designer begint met het ontwerpen van de user interface, waarna de ontwikkelaar hier gedrag en data aan koppelt.

In principe zijn er twee manieren waarop je dit kan aanpakken. In de eerste plaats zou je de user interface kunnen ontwerpen in Expression Blend, waarna je in Visual Studio door middel van data-binding de koppeling met het model realiseert. Dit werkt prima als je alleen werkt, maar als je samenwerkt met een designer dan ga je elkaar al snel in de weg zitten. Zo kan het gebeuren dat je te maken krijgt met merge-conflicten omdat de designer iets heeft toegevoegd aan een XAML bestand, dat jij ondertussen als ontwikkelaar ook hebt zitten aanpassen. Daarnaast kan de designer in Expression Blend niet zien hoe het resultaat zal zijn met de data gekoppeld aan de user interface.

De tweede methode is om voorbeelddata te definiëren in Expression Blend. Op deze manier kun je zien hoe de user interface eruit ziet met gegevens eraan gekoppeld. Als je samenwerkt met een designer heeft dit nog een voordeel, namelijk dat je elkaar niet



FIGUUR 1: MENUSTRUCTUUR NIEUWE SAMPLE DATASOURCE.



FIGUUR 2: NEW SAMPLE DATA DIALOOG.

langer in de weg zit. De designer kan zich bezighouden met het ontwerp in Expression Blend, terwijl jij je kan richten op het implementeren van het model.

Om te kunnen werken met voorbeeldgegevens in Expression Blend 4 hoef je niet perse gebruik te maken van een bestaand model. Je kan ook zelf voorbeeldgegevens maken, door middel van het "Create sample data" menu op het data paneel in Expression Blend.

In Figuur 1 zie je hoe dit menu eruit ziet. Middels het "Create sample data menu" kun je een sample datasource maken op basis van een bestaand model of een XML bestand importeren en op basis daarvan een nieuwe sample datasource maken. Ook kun je er voor kiezen om met de hand een datasource samen te stellen. Kies om een nieuwe sample datasource te maken voor de optie Create Sample data. Er verschijnt nu het dialoogvenster zoals in figuur 2 is weergegeven.

Je kan in dit scherm een aantal opties configureren voor de nieuwe datasource. Allereerst kun je de datasource een naam geven, deze naam vind je later terug in het data paneel. Ook kun je in dit scherm middels de 'Enable sample data when the application is running' optie aangeven of het een designtime datasource betreft of een runtime datasource. Tot slot kun je nog aangeven of de datasource in het project moet worden opgenomen, of dat de datasource alleen beschikbaar dient te zijn voor het scherm dat je op dat moment hebt geopend in de designer.

Het al dan niet uitschakelen van het tonen van voorbeeld data tijdens runtime heeft invloed op de manier waarop de datasource wordt gekoppeld. Laat je het vinkje aanstaan, dan zal bij het binden van de gegevens in Blend de DataContext property worden gebruikt om de datasource te koppelen. Dit heeft tot gevolg dat de voorbeelddata zal worden getoond als je de applicatie start. Overschrijf je de DataContext property in de constructor of tijdens het afhandelen van het Load event van het scherm, dan zul je merken dat je in Blend gewoon gebruik kan blijven maken van de voorbeelddata. Werk je niet op deze manier, maar koppel je het model in XAML, dan moet je kiezen tussen het model of de voorbeelddata. Laat je het vinkje uitstaan, dan wordt de DataContext property uit de blend designtime XML namespace gevuld. Omdat deze namespace tijdens runtime wordt genegeerd, zal er geen data worden getoond wanneer je de applicatie start. Bij deze laatste methode heb je als ontwikkelaar echter wel volledige vrijheid en maakt het niet uit op welke manier je het echte model koppelt.

## Properties toevoegen

Zodra je de sample datasource hebt aangemaakt verschijnt deze in het Data paneel. Een voorbeeld hiervan is weergegeven in figuur 3. Je kan in het data paneel vervolgens de sample datasource bewerken door hier properties aan toe te voegen of door de bestaande properties te bewerken.

Om een property toe te voegen klik je op de Add simple property optie achter de naam van de sample datasource. Nadat je de property hebt toegevoegd kun je de naam hiervan wijzigen door dubbel te klikken op de property. Daarnaast kun je ook de metadata van de property wijzigen door op de dropdown achter de property te klikken. In dit menu kun je aangeven wat het datatype van de property is, daarbij kun je kiezen uit number, boolean, image of string. Ook kun je aangeven hoe de voorbeelddata voor deze property eruit moet zien.

Als je een property niet langer nodig hebt kun je deze verwijderen uit de datasource door rechts te klikken op deze property en te kiezen voor de optie Remove property. De property wordt dan verwijderd uit de sample datasource. Eventuele bindings die verwijzen naar de property blijven echter wel bestaan, het is dan ook belangrijk om even na te kijken of de property nog ergens is gekoppeld in de user interface. Je kan eventuele problemen met bindings tijdens het uitvoeren van de applicatie zien in de console output binnen zowel Expression Blend als Visual Studio.

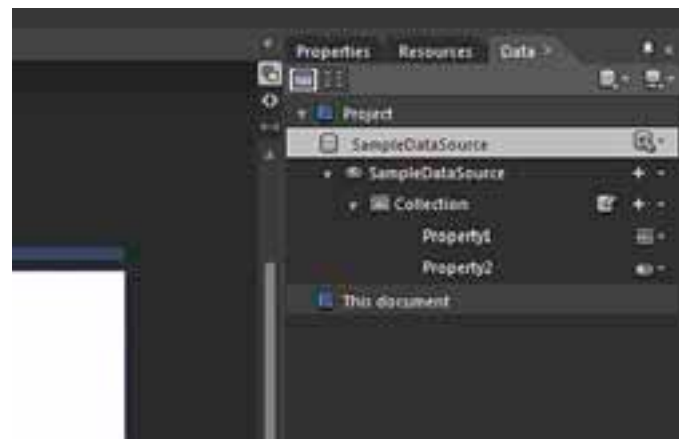
## Collecties toevoegen

Naast properties is het ook mogelijk om collecties van items te definiëren. Deze kun je gebruiken om bijvoorbeeld een DataGrid of een ListBox control te vullen met een aantal voorbeelditems. Om een collectie toe te voegen aan een sample datasource klik je op de dropdown achter de Add simple property knop op de datasource en kies je voor de optie Add collection property. Er verschijnt nu een nieuwe collectie in de sample datasource. Nadat je de collectie hebt toegevoegd kun je met de Add simple property eigenschappen toevoegen aan de items in de collectie of een subcollectie definiëren. Door collecties en eigenschappen te combineren kun je complexere datastructuren modelleren die je op een later moment kan databinden aan de user interface.

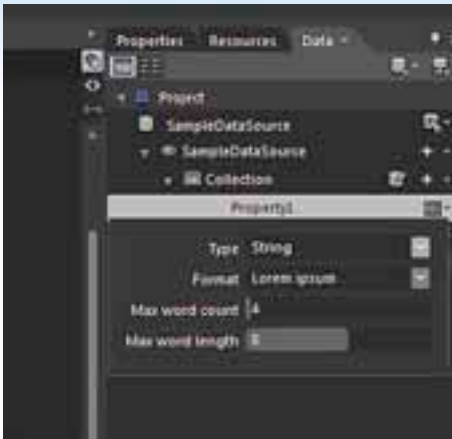
## Hiërarchische gegevens tonen

Wanneer je een collectie hebt gedefinieerd kun je deze omzetten naar een hiërarchische structuur. Deze structuur kun je gebruiken om bijvoorbeeld een TreeView control te vullen met gegevens. Het omzetten van een collectie naar een boomstructuur doe je door op de dropdown achter de collectie te klikken en te kiezen voor de optie Convert to Hierarchical collection. Dit zorgt ervoor dat er een boomstructuur wordt opgebouwd. Een voorbeeld hiervan zie je in figuur 4.

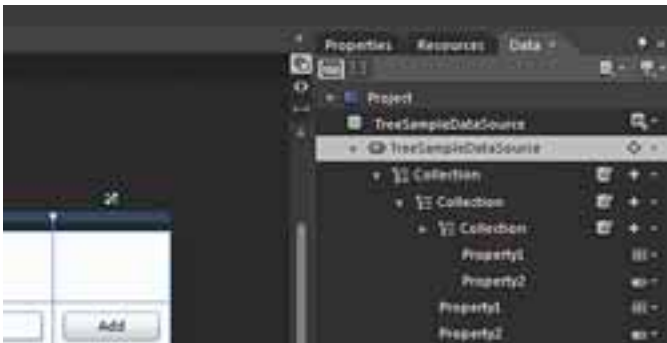
De eigenschappen van de items in de boomstructuur kun je achteraf wijzigen. Deze wijzigingen worden vervolgens automatisch



FIGUUR 3: WEERGAVE SAMPLE DATASOURCE IN HET DATA PANEEL.



FIGUUR 4: WEERGAVE DATASOURCE MET HIERARCHISCHE STRUCTUUR.



FIGUUR 5: PROPERTY METADATA

toegepast op alle niveaus in de boomstructuur. Het gebruik van een boomstructuur werkt het beste wanneer je één soort items op alle niveaus in de boomstructuur wilt tonen. Wil je verschillende soorten items op verschillende niveaus in de boomstructuur tonen dan kun je het beste gebruik maken van collecties met subcollecties.

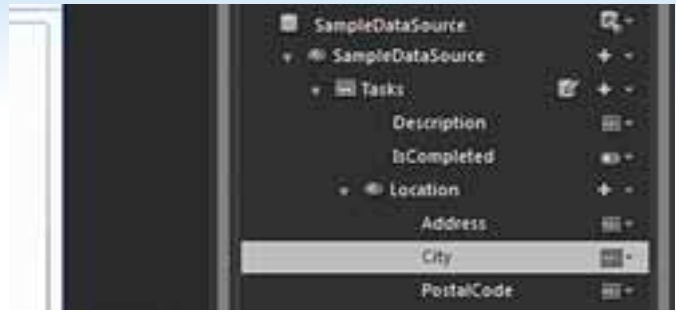
## Complex properties

Naast simple properties, kun je er ook voor kiezen om zogenaamde complex properties toe te voegen. Een complex property kan het beste worden vergeleken met een class. Op deze manier kun je bijvoorbeeld een structuur maken met een Persoon met daar een Adres object aan gekoppeld. Aan een complex property kunnen zowel simple properties als complex properties en collecties worden toegevoegd.

## De gegenereerde gegevens bewerken

Standaard wordt voor een datasource door Expression Blend een set voorbeeldrecords gegenereerd. Deze gegevens zijn over het algemeen prima toepasbaar om de user interface te ontwerpen. Wil je echter andere data tonen dan wat standaard is gegenereerd dan kan dat ook.

De eerste manier om de voorbeelddata aan te passen is door de metadata van properties aan te passen. De metadata van een property op een datasource kun je bewerken door op de dropdown



FIGUUR 6: DIALOGOVENSTER VOOR HET BEWERKEN VAN VOORBEELDDATA.

achter de property te klikken. In figuur 5 zie je hier een voorbeeld van. Bij de number, boolean en image property kun je de voorbeelddata niet heel uitgebreid aanpassen, maar bij een string property kun je wel aangeven wat voor soort voorbeelddata er moet worden gegenereerd. Zo kun je er voor kiezen om standaard lorem ipsum te laten genereren, maar bijvoorbeeld ook voor e-mailadressen of url's. In totaal zijn er 11 verschillende generatoren waaruit je kan kiezen. Het is niet mogelijk om zelf generatoren te schrijven.

Voor volledige controle kun je tot slot op de Edit sample data knop klikken achter een collectie. In het edit sample values dialogovenster kun je allereerst aangeven hoeveel records je wilt genereren, vervolgens kun je hier ook de gegevens voor de individuele records wijzigen. In figuur 6 is het "Edit Sample Values" dialogovenster weergegeven.

Alle wijzigingen die je aanbrengt in de voorbeelddata worden opgeslagen in een apart bestand binnen het project, zodat je ze op een later moment kan weer opnieuw kan gebruiken. Het prettige hieraan is ook dat dezelfde voorbeelddata ook in Visual Studio 2010 kan worden gebruikt.

## XML gegevensbestanden gebruiken

Een andere manier waarop je een sample datasource kan definiëren, zonder dat je een bestaand model hebt, is door gebruik te maken van XML bestanden. Deze methode is bijvoorbeeld handig wanneer je op basis van een XML schema een aantal voorbeeldbestanden hebt gegenereerd en deze wilt gebruiken om op basis hiervan de user interface te ontwerpen.

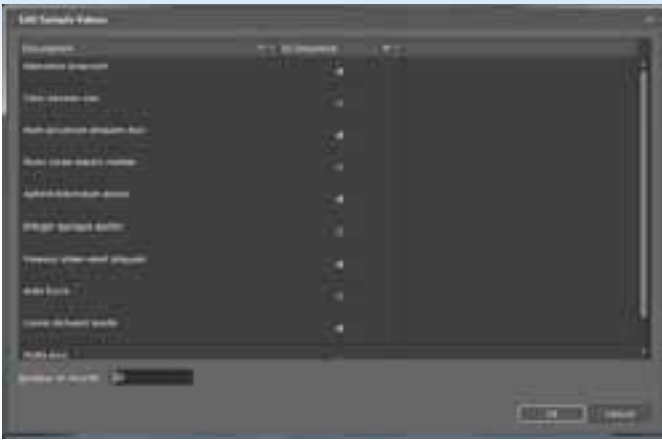
Bij het importeren zal Expression Blend het XML bestand inlezen en hiervoor zelf een structuur bedenken. Deze structuur zal in veel gevallen overeenkomen met de structuur in het XML bestand. Je zult echter merken dat als je collecties hebt gedefinieerd in het XML bestand dat de naamgeving hiervan niet altijd overeenkomt met wat je had willen zien. Zo zal er bijvoorbeeld voor een set Adres elementen een collectie worden aangemaakt in de datasource met de naam AdresCollection, terwijl je misschien liever Adressen als naam had gewild.

Nadat je het bestand hebt geïmporteerd zal de sample datasource in het data paneel verschijnen en kun je de data uit de sample datasource koppelen aan de user interface in de designer.

---

Bij het importeren zal Expression Blend het XML bestand inlezen en hiervoor zelf een structuur bedenken, maar de naamgeving komt niet altijd overeen.

---



FIGUUR 7: DATABINDING DIALOOGVENSTER

## Databinding met voorbeeldgegevens

Als je eenmaal een sample datasource met een set voorbeelddata hebt toegevoegd aan het project, kun je deze data koppelen aan de user interface elementen in de designer door properties of collecties uit het data paneel op de designer te slepen.

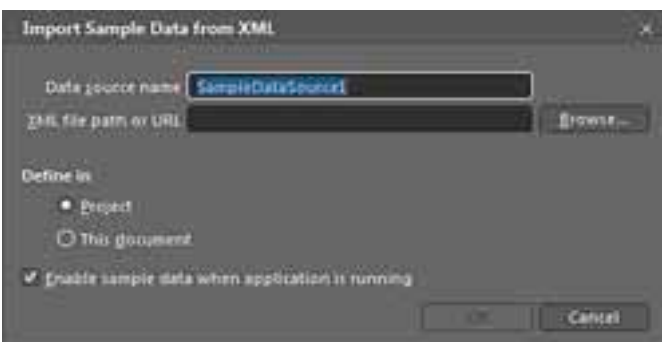
Het slepen van properties op de designer is niet de enige manier waarop je gegevens uit een sample datasource kan databinden. Als je op het kleine vierkantje achter een property van een control klikt op het properties paneel kun je kiezen voor de optie Databinding waarna je een dialoogvenster zoals in figuur 7 te zien krijgt.

In dit dialoogvenster kun je niet alleen de property aangeven die je wilt koppelen, maar ook opties als de richting van de binding aangeven en een converter kiezen die je wilt gebruiken om de gegevens die worden gekoppeld te converteren naar het gewenste formaat. Door het gebruik van voorbeeld data hoeft je dus niet de naam van de te koppelen eigenschap te raden. Het dialoogvenster toont alle beschikbare velden en voorkomt op die manier fouten bij het koppelen van velden. Je zult ook merken dat je sneller in staat bent om een werkende user interface op te leveren.

## Werken met een bestaand model

Hiervoor heb ik laten zien hoe je zonder bestaande code een user interface kan ontwerpen met behulp van voorbeelddata. Heb je wel een model, dan kun je nog steeds gebruik maken van deze feature. Expression Blend ondersteunt namelijk het maken van sample datasources op basis van bestaande classes.

Wanneer je een nieuwe datasource op basis van een bestaande class toevoegt, zul je het dialoogvenster uit Figuur 8 te zien krij-



FIGUUR 8: DIALOOGVENSTER VOOR HET MAKEN VAN EEN SAMPLE DATASOURCE OP BASIS VAN BESTAANDE CODE.

gen. In dit dialoogvenster kun je een keuze maken uit de verschillende classes die beschikbaar zijn voor de applicatie. Soms worden hier niet alle classes getoond, in dat geval kun je het beste nog een keer het project bouwen en nogmaals de datasource toevoegen. Expression Blend toont namelijk alleen gecompileerde classes in dit dialoogvenster. Heb je veel classes dan kun je in dit dialoogvenster ook een deel van de naam van de class opgeven, de lijst zal in dat geval gefilterd worden op deze term.

Nadat je de sample datasource hebt aangemaakt kun je de data die is gegenereerd voor datasource bewerken. Er is wel een belangrijk verschil met een sample datasource die niet op basis van een bestaande class is gemaakt. Wanneer je op de *Edit sample data* knop achter de datasource klikt krijg je namelijk het XAML bestand te zien waarin de sample datasource is gedefinieerd. Er is geen editor meegeleverd om de waardes te bewerken.

Het bijzondere is dan wel weer dat Microsoft er aan gedacht heeft om bij properties op de sample datasource het mogelijk te maken om de waarde voor properties anders te laten genereren. Hierbij kun je in het geval van string properties kiezen uit de eerder genoemde generators voor het genereren van voorbeeldwaardes, zoals de e-mailadres generator en de lorem ipsum generator.

## Conclusie

De mogelijkheid om gebruik te maken van voorbeeldgegevens in Expression Blend maakt het niet alleen eenvoudiger om een user interface te ontwerpen. Het is een essentieel hulpmiddel om de samenwerking tussen ontwikkelaars en designers soepel te laten verlopen. Het zorgt ervoor dat je elkaar minder in de weg zit.

Combineer het MVVM design-pattern met voorbeeldgegevens en je zult merken dat het nog eenvoudiger wordt om het ontwerp te ontkoppelen van het gedrag van de user interface. Je kan als designer gewoon doorwerken aan het ontwerp, terwijl je als ontwikkelaar weer prima kan werken aan het ontwerpen en unit-testen van het viewmodel en de koppeling met eventuele webservices en dergelijke.



.....  
**Willem Meints**, is consultant bij Info Support voor de business unit Zorg. Naast zijn werk is hij een actief blogger waarbij hij zich voornamelijk bezighoudt met Silverlight en frontend ontwikkeling in het algemeen.

Bronkhorst High-Tech is een modern en innovatief bedrijf. Het houdt zich bezig met ontwikkeling, productie en verkoop van zeer hoogwaardige meet- en regelinstrumenten voor gas- en vloeistofstromen. Vanuit haar vestigingsplaats Ruurlo heeft Bronkhorst zich ontwikkeld tot Europees marktleider op het gebied van Mass Flow Controlles.



De afdeling Informatie- en communicatietechnologie bij Bronkhorst High-Tech B.V. bestaat uit 3 groepen: ICT-development, ICT-security en ICT-operations. De software ontwikkelaars in de ICT-development groep zorgen samen met de afdelingen Research & Development en Engineering, vanuit de vestiging in Ruurlo, voor de innovatie en ontwikkeling van nieuwe producten. De nieuwe ideeën, mede door de markt gevoed, worden door het uitvoeren van haalbaarheidsstudies onderzocht. Hierna volgt de productontwikkelingsfase die door een projectteam, samengesteld uit medewerkers van de diverse disciplines en afdelingen wordt uitgevoerd. Om de groep ICT-development van Bronkhorst High-Tech te versterken zijn wij op zoek naar een:

## Software ontwikkelaar .NET (m/v)

Als Software ontwikkelaar ontwerp en ontwikkel je in Microsoft .NET applicaties ten behoeve van onze producten en processen.

### Uw loopbaan:

- ▶ u doet kennis en ervaring op in een groeiend, innovatief en dynamisch bedrijf;
- ▶ u werkt aan één of meerdere technisch multidisciplinaire projecten, gericht op innovatie, ontwikkeling en verbetering van nieuwe en bestaande producten en/of processen;
- ▶ uw werkzaamheden bestaan o.a. uit: het opzetten van een functioneel ontwerp, technisch ontwerp, programmacode schrijven, testen en het documenteren op een agile gebaseerde werkwijze

### Uw profiel:

- ▶ een ervaren Microsoft .NET specialist;
- ▶ in het bezit van een HBO diploma;
- ▶ aantoonbare kennis en ervaring met WCF, Ajax, WPF en/of WF;
- ▶ snel in staat nieuwe technologieën in te zetten;
- ▶ een sterk analytisch denkvermogen;
- ▶ teamplayer met een flexibele, pragmatische en resultaat gerichte instelling;
- ▶ uitstekende communicatieve en contactuele vaardigheden;
- ▶ creativiteit en doorzettingsvermogen;
- ▶ goede beheersing van Engelse taal;
- ▶ kennis van SQL Server en/of Visual Basic 6 is een pré;
- ▶ het bezit van een relevante MCSD / MCTS certificering is een pré;

### Interesse?

Uw schriftelijke sollicitatie kunt U richten aan:  
Bronkhorst High-Tech B.V.  
t.a.v. mevr. I. Mensink,  
Nijverheidsstraat 1A, 7261 AK Ruurlo;  
of [sollicitaties@bronkhorst.com](mailto:sollicitaties@bronkhorst.com)

Voor meer informatie kunt u contact opnemen met mevr. I. Mensink, tel. (0573) 45 88 00. Meer informatie over ons bedrijf kunt u vinden op onze website: [www.bronkhorst.com](http://www.bronkhorst.com).