

Een datagedreven onderhoudsomgeving

Bert Dingemans
Februari 2015

Inhoudsopgave

Waarom een onderhoudsomgeving voor (master) data?	3
Enkelvoudig beheer en meervoudig gebruik	3
Hergebruik van software	3
Reductie van complexiteit	3
Wat is een onderhoudsomgeving?	4
Definitie onderhoudsomgeving (master) data	4
Ontsluiten van applicatie-, register- - en/of master data	4
Hergebruik door standaardisatie	5
Onderhoudsomgeving inrichting	5
Algemene requirements voor de onderhoudsomgeving	5
Specifieke requirements	5
Overzicht van de onderhoudsomgeving	6
Dynamic data	7
Visual Lightswitch	7
Business Connectivity Services	8
Sharepoint Access Services	8
Ontwikkelproces	8
Processtappen	9
Rollen ¹⁰	
Criteria inzet onderhoudsomgeving	11
Voorbeeld implementatie	12
Bijlage scorematrix tooling	13

Veel organisaties hebben een applicatielandschap waarbij de introductie van maatwerkoplossingen wordt ontmoedigd en de introductie van hergebruik of het aanschaffen van standaardpakketten de voorkeur heeft. Daarnaast is er een trend dat er master data sets ingezet worden die op meerdere plekken in de organisatie worden gebruikt in werkprocessen. Ongewenste ontwikkeling is dat het beheer van deze (master) gegevenssets wordt ondergebracht in specifieke applicaties in plaats van in generieke voorzieningen.

Omdat de inzet van generieke voorzieningen vanuit kosten- en beheeroogpunt de voorkeur heeft ten opzichte van ontwikkeling van maatwerkapplicaties is de inzet van een generieke voorziening voor het beheer van (master) data een interessant scenario. Dit document beschrijft een mogelijke implementatie van een dergelijke onderhoudsomgeving voor data.

Waarom een onderhoudsomgeving voor (master) data?

Het introduceren van een onderhoudsomgeving voor master data wordt gedaan vanuit een aantal beweegredenen. De belangrijkste worden hieronder uitgewerkt gerelateerd aan een aantal data architectuur principes

Enkelvoudig beheer en meervoudig gebruik

Het beheer van data kent verschillende niveaus van kwaliteit. Voor herbruikbare en generieke gegevens worden registers ingezet zoals een master data omgeving, authentieke registraties of kernregisters. Voor complexe bedrijfsprocessen worden veelal materiestructuren ingezet zoals ERP, DMS of CMS systemen. Daarnaast zijn er gegevens voor eigen gebruik van medewerkers met een geringe complexiteit, denk aan excel lijsten e.d.

Tussen gegevens in applicaties (materiestructuren) en bestanden voor eigen gebruik zijn generieke gegevenssets te onderkennen. Het onderhouden en beheren van deze gegevenssets dient op één plek in de organisatie belegd te worden (veelal bij functioneel beheer) waarbij het vervolgens gebruikt kan worden door meerdere afnemers.

Vindt het beheer plaats door niet ITers dan worden er aan de gebruikersinterface en –navigatie andere eisen gesteld dan bij beheer door ITers. Deze onderhoudsomgeving voorziet in een eenvoudige maar gebruikersvriendelijke gebruikersinterface die relatief eenvoudig geïmplementeerd kan worden.

Hergebruik van software

Voor software geldt de selectievolgorde hergebruik – kopen – maatwerk. Inzetten van een generieke omgeving die grotendeels geconfigureerd kan worden zonder de toevoeging van maatwerk valt dan onder hergebruik. De Inzet van deze onderhoudsomgeving is gebaseerd op generieke componenten waarbij maatwerk beperkt blijft tot het configureren van de gebruikersinterface. Inrichting van de onderhoudsomgeving met generieke functionaliteit valt daardoor ook in de categorie hergebruik.

Reductie van complexiteit

Door het ontbreken van een onderhoudsomgeving voor (master) data zie je een tendens om specifieke applicaties te gaan ontwikkelen voor het beheer (master) data of bestaande applicaties aan

te passen zodat deze geschikt worden gemaakt voor het onderhoud van (master) data. Veelal wordt hierdoor de configuratie van bestaande applicaties complexer. Dit omdat deze niet ingericht zijn op het configureren van navigatie en gebruikersinteractie.

Door inzet van een generieke voorziening voor ontsluiting en beheer van (master) data kan complexiteit gereduceerd worden. Dit maakt introductie van nieuwe ontsluitingen eenvoudiger en zal op termijn een reductie van beheerkosten met zich mee brengen.

Wat is een onderhoudsomgeving?

In het vorige hoofdstuk is reeds ingegaan op een aantal knelpunten bij onderhoud, beheer en gebruik van (master) data. Uitgaande van het principe “**enkelvoudig beheer en meervoudig gebruik**” kan gezocht worden naar een combinatie van een gewijzigde inrichting van de ICT voorzieningen en inrichting van beheerprocessen. Dit hoofdstuk gaat met name in op een wijziging in de ICT inrichting. In volgende hoofdstukken wordt ingegaan op ontwikkel- en beheerprocessen.

Definitie onderhoudsomgeving (master) data

Een onderhoudsomgeving voor (referentie) data kan ingezet worden bij de volgende gebruikskennmerken:

- Gegevens die opgeslagen worden in een relationele database
- Gegevenssets met een relatief eenvoudig relationeel datamodel
- Ontbreken van complexe gegevensvalidaties.
- Gegevenssets die gecombineerd cq uitgebreid worden met register- en/of applicatiedata
- Gebaseerd op een eenvoudig beheer- of werkproces
- Veelal kleine gebruikerspopulatie
- Gebruik bestaat voornamelijk uit beheerwerkzaamheden

Om een beeld te vormen van referentie data een aantal voorbeelden:

- Eenvoudige (code)lijsten die door functioneel beheerders of eindgebruikers beheerd worden (denk aan product.of artikelcatalogi e.d.)
- Enkelvoudige data bijvoorbeeld de opslag van eenvoudige e-formulier gegevens waarop vervolg bewerkingen uitgevoerd moeten worden
- Verrijkingen van registerdata die eenvoudig beheer vragen. Bijvoorbeeld het combineren van data uit twee verschillende materiestructuren (zoals medewerkergegevens koppelen aan regio's).

Ontsluiten van applicatie-, register- - en/of master data

Naast master data onderkennen we applicatie data, veelal gegevens ter ondersteuning van een bedrijfsproces. Daarnaast wordt registerdata onderkend. Dit zijn gegevens in materiestructuren die procesoverstijgend gebruikt worden. Denk hierbij aan relatie- of medewerkergegevens.

Master data kan gecombineerd worden met deze applicatie- of registerdata waardoor er waarde toegevoegd wordt aan de totale set van data. Dit houdt in dat deze verrijking een verbijzondering is van deze data zonder dat deze verrijking generiek is. Door het combineren van deze gegevenstypen in één onderhoudsomgeving kan de complexiteit van het applicatielandschap gereduceerd worden.

Hergebruik door standaardisatie

Bij de inrichting van de master data omgeving is het mogelijk om op verschillende vlakken te standaardiseren waardoor hergebruik mogelijk wordt. Bijvoorbeeld:

- Inzet van hetzelfde databaseplatform als bijvoorbeeld het datawarehouse gegevensvoorziening
- Gebruik maken van (open source) componenten voor gegevensontsluiting zoals Entity Framework en Dynamic Data
- Inzet van bestaande generieke voorzieningen zoals:
 - CMS voor portaal en autorisatie functionaliteit
 - Eformulier omgeving voor het ontwikkelen van gebruikersvriendelijke E-formulieren voor externe gebruikers
 - Tooling voor eventuele rapportages
 - Ontwikkelplatform van webservices voor applicatie integratie
- Gebruik van algemene huisstijl voor ondersteuning van de webrichtlijnen

Onderhoudsomgeving inrichting

Algemene requirements voor de onderhoudsomgeving

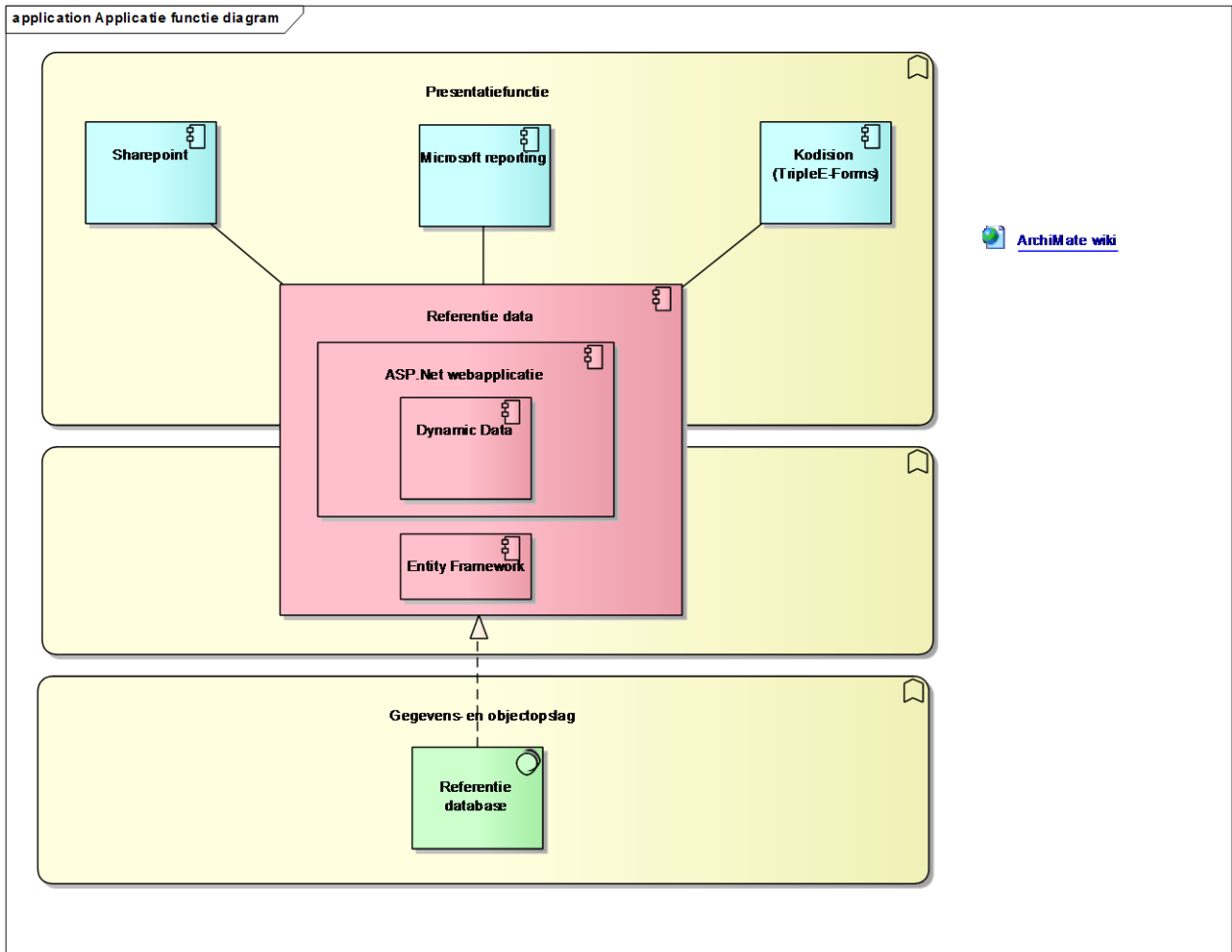
Aan een onderhoudsomgeving worden de volgende requirements gesteld

- **Datamodel en beheer van de User-Interface alleen configureren**, inrichten van een nieuwe onderhoudsomgeving maakt gebruik van de generieke inrichting waarbij het datamodel en de user-interface via modelleertooling gegenereerd worden
- **Geen maatwerk (niet programmeren)**, de generieke omgeving kent eventueel geprogrammeerde functionaliteit echter voor een specifieke inrichting is maatwerk niet toegestaan behalve de inzet van de hieronder genoemde componenten
- **Webbased oplossing**, het dient een webbase oplossing te zijn te gebruiken in de gangbare browsers.
- **Open source software**, bij voorkeur (extra) componenten inzetten vanuit CodePlex ter voorkoming van een vendor lock in.

Specifieke requirements

- **Uitwerking beperking op basis van Microsoft componenten.**
 - WCF voor webservices
 - SQL server database
 - Sharepoint voor autorisatie- en portaalfunctionaliteit
 - Reporting services voor rapportages

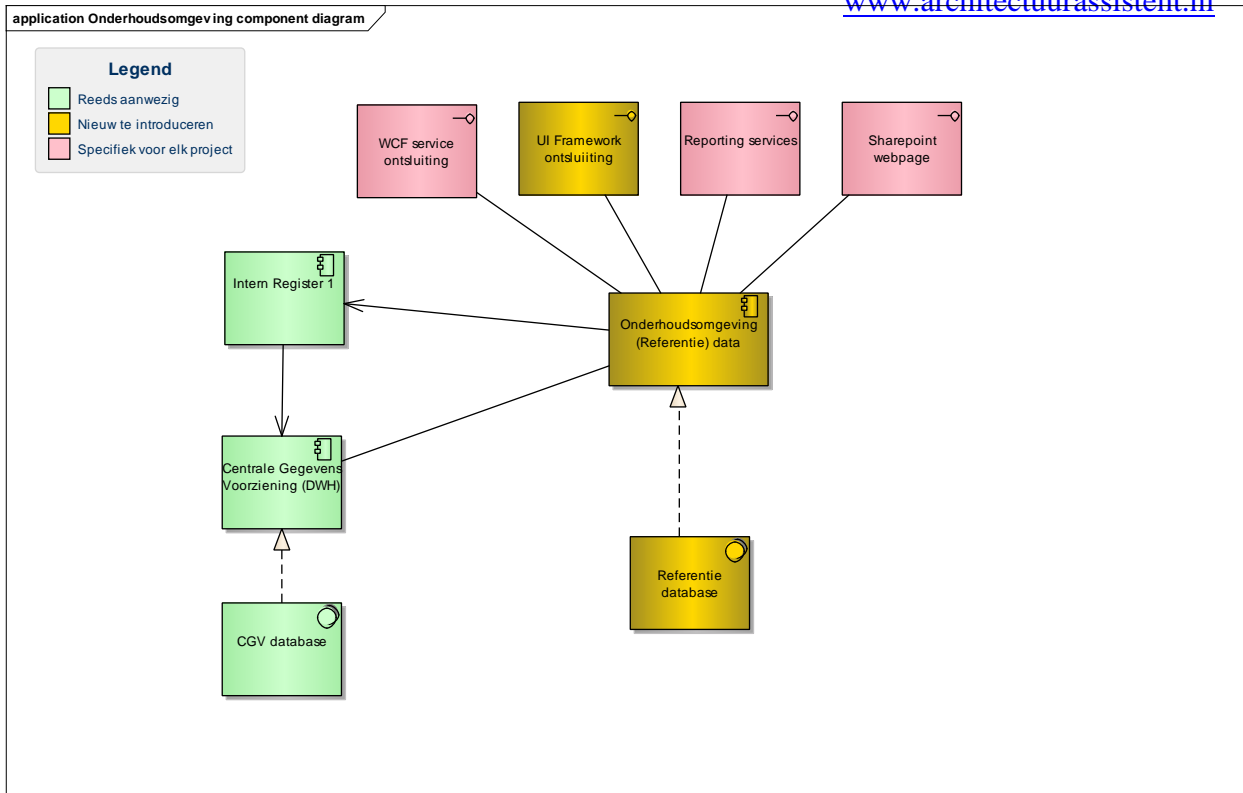
Overzicht van de onderhoudsomgeving



In bovenstaande afbeelding zie je de verschillende componenten afgebeeld op de logische applicatielagen van een standaard applicatie. Omdat de opzet van de applicatie eenvoudig is zie je dat de bedrijfslaag verhoudingsgewijs eenvoudig is. In het voorbeeld wordt uitgegaan van het dynamic data model, echter ook voor de andere inrichtingswijzen zoals lightswitch zal een soortgelijke uitwerking gelden.

Naast een uitwerking van het logische model is een componenten en gegevensstromen diagram opgesteld. In onderstaande figuur is deze uitgewerkt. Dit diagram heeft de volgende kenmerken:

- Hergebruik van bestaande componenten zoals Sharepoint en WCF
- Integratie met bestaande voorzieningen als interne registers



Voor bovenstaande uitwerking zijn een aantal implementatiescenario's op basis van Microsoft technologie mogelijk, deze worden hieronder uitgewerkt en met elkaar vergeleken.

Dynamic data

Dynamic Data is een open source oplossing binnen het ASP.Net framework. Het wordt gebruikt in combinatie met het Entity Framework van Microsoft. Het is aanwezig als een template in visual studio. De initiële inrichting vergt minimaal programmeerwerk, de latere inrichting nauwelijks mits geen gebruik gemaakt wordt van scaffolding.

Voordelen	Nadelen
<ul style="list-style-type: none"> Goedkoop Voldoet aan architectuureisen Aanpasbaar en uitbreidbaar Sluit aan bij reeds gebruikte modules zoals Entity Framework OTAP inrichting eenvoudig en script gebaseerde updates 	<ul style="list-style-type: none"> Toekomstvastheid Bepaalde configureerfunctionaliteit snel maatwerk gewenst

Visual Lightswitch

Dit is een product dat alleen beschikbaar is in de professional versie van Visual Studio. Het bestaat uit een gebruikersvriendelijke interface voor datamodellering en schermontwerp. Er is een webbased en desktop versie beschikbaar

Voordelen	Nadelen
<ul style="list-style-type: none"> Voldoet aan architectuureisen Toekomstvast Aanpasbaar en uitbreidbaar Integratie met Sharepoint 	<ul style="list-style-type: none"> Schermen moeten altijd geconfigureerd worden Introductie eigen templates beperkt mogelijk

<ul style="list-style-type: none">• Mobile Apps• Goedkoop• OTAP inrichting eenvoudig	<ul style="list-style-type: none">• Knelpunt/Bug in de huidige configuratie bij het gebruik van views
--	---

Business Connectivity Services

Is een onderdeel van sharepoint voor het ontsluiten van bedrijfsdata. Het biedt meerdere typen servicelagen voor ontsluiting van de gegevens, waaronder een rest interface. Wanneer deze ingezet wordt zal snel maatwerk ingezet dienen te worden

Voordelen	Nadelen
<ul style="list-style-type: none">• Toekomstvast• Hergebruik sharepoint functionaliteit	<ul style="list-style-type: none">• Niet geschikt voor complexere oplossing• Duur• OTAP inrichting is complex

Sharepoint Access Services

Op basis van een Access client kunnen applicaties gebouwd worden die vervolgens in Sharepoint gepubliceerd worden. Het is een krachtig mechanisme maar vergt wel vrij snel maatwerk aanpassingen

Voordelen	Nadelen
<ul style="list-style-type: none">• Toekomstvast• Hergebruik sharepoint functionaliteit	<ul style="list-style-type: none">• Maatwerk• Duur• OTAP inrichting is complex?

Bij de verdere uitwerking in een proeftuin zijn een aantal platformen met elkaar vergeleken en is een scorematrix opgesteld. Deze is te vinden in de bijlage

Ontwikkelproces

Het ontwikkelen van een onderhoudsomgeving bestaat uit een aantal deelactiviteiten met focus op de inrichting van een aantal onderdelen:

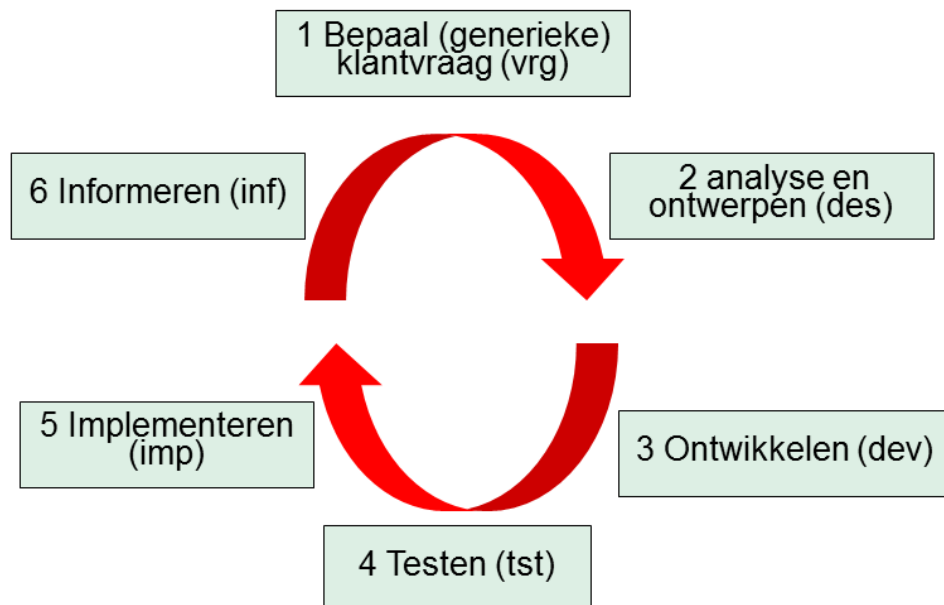
- Ontwerpen van het data model op basis van database first of model first
- Inrichten van de databaselaag binnen de database inclusief het gebruik van views voor generieke gegevenssets
- Ontsluiten van de databaselaag via het entity framework
- Eventueel ontwikkelen/configureren van andere componenten zoals rapporten, webservices of autorisaties.

Daarnaast kenmerkt het ontwikkelproces van een onderhoudsomgeving implementatie zich door een iteratief karakter. Opzet is dat bij de initiële inrichting er een eenvoudig datamodel gemaakt wordt en dat bij latere gebruikerswensen het datamodel wordt aangepast of uitgebreid. Is dit het geval dan zal een nieuwe iteratie opgestart worden. Hierbij is echter het aantal iteraties beperkt tot maximaal drie. Daarna wordt via wijzigingsbeheer en releases zorggedragen voor een acceptabele ontwikkel inspanning van de ontwikkelafdeling/leverancier. Bij deze uitwerking is het essentieel dat het datamodel vooraf bepaald is.

Processtappen

In onderstaande afbeelding de stappen van een iteratie bij inzet van de onderhoudsomgeving. Het iteratief ontwikkelen op basis van prototyping is in deze een goede te zetten werkwijze. Vandaar dat in onderstaande uitwerking dit als uitgangspunt geldt. Daarnaast is een iteratie gericht op het uitwerken van een datamodel en vervolgens het configureren van de onderhoudsomgeving. Ontwikkelactiviteiten op basis van programmeren van broncode is niet wenselijk, in die situatie moet gekozen worden voor een ASP.Net maatwerk ontwikkeltraject.

Ontwikkel- en beheerproces



De processtappen zijn gebaseerd op het ASL model en zijn iteratief ingericht. De stappen hebben de volgende beschrijving:

- **Bepaal klantvraag**, bij het uitwerken van het model van de gegevensontsluiting moet gekeken worden naar de behoefte van de vraagsteller en in hoeverre een referentie data omgeving ingezet kan worden. Hierbij wordt allereerst een datamodel opgesteld. Tevens wordt gekeken welke data-entiteiten in dit model uit registers en applicaties gebruikt kunnen worden. Als laatste dient bepaald te worden welke aanvullende onderdelen ingezet moeten worden, zoals:
 - **WCF voor webservice ontsluiting**
 - **Sharepoint voor portaal- en autorisatie functie**
 - **Reporting services voor rapportages**
 - **Eventueel e-formulieren**
- **Analyse en ontwerpen**, opstellen van een ontwerpdocument met een functionele beschrijving van het datamodel. Hierbij wordt gebruik gemaakt van een sjabloon (zie de sharepoint projectsite).
- **Ontwikkelen**, is het ontwerp akkoord dan wordt het datamodel ontwikkeld in de database en wordt een model opgesteld in het Entity Framework. Indien gewenst wordt de scaffolding uitgewerkt en worden de aanvullende componenten ingericht

- **Testen**, tijdens de testfase zullen de systeem- en acceptatietest uitgevoerd worden. Na acceptatie wordt de referentie data omgeving geïmplementeerd
- **Implementeren**, dit is het overzetten van de inrichting van de OT omgeving naar de productieomgeving van de Referentie Data inrichting.
- **Informer**, Betrokken informeren over de introductie en zorgdragen dat het register met referentie data en de registers met services en generieke data elementen is bijgewerkt

Rollen

Bij ontwikkeltrajecten zijn de volgende rollen betrokken:

- **Eigenaar/Afnemer (Eig)**, bewaakt de vraagstelling. Is verantwoordelijk voor de uiteindelijke inrichting van de gegevensontsluiting
- **Projectleider (Prj)**, draagt zorg voor het op tijd en binnen de gestelde resources van de gewenste referentie data
- **Informatieanalyse (IA)**, beschrijft de informatiebehoefte van de afnemers en vertaalt deze in een specificatie van het datamodel en de ontsluitingscomponenten
- **Data ontwerp (DO)**, beschrijft de gegevensdefinitie en -implementatie op basis bovengenoemde informatiebehoefte
- **Data realisatie (DR)**, reasiseert het datamodel binnen de database en in het Entity Framework op basis van bovengenoemde analyse en ontwerp
- **Architectuur (Arc)**, stelt kaders aan implementaties en besluit over de inzet van maatwerk op basis van belangen van de diverse stakeholders
- **Tester (Tst)**, test de gehele implementatie van de referentie data in opdracht van de eigenaar en de afnemer(s)
- **Beheerder (Bhr)**, neemt de ontwikkelde omgeving in beheer, draagt zorg voor de continuïteit en signaleert trends in het gebruik.

Deze rollen worden veelal gecombineerd en uitgevoerd door één actor. Zo kunnen bijvoorbeeld de architectuur, informatie analyse en ontwerp rollen eenvoudig gecombineerd worden en uitgevoerd worden door één actor.

In onderstaande matrix worden de procestappen gerelateerd aan de rollen:

	Eigenaar	Projectleider	Informatie Analist	Data Ontwerp	Data Realisatie	Architectuur	Testen	Beheerder
Vraag	X	X	X			X	X	
Ontwerp		X	X	X		X	X	X
Ontwikkeling		X			X			?
Test	X	X			X			X

Implementatie		X			X			
Informereren	X	X				X		X

Criteria inzet onderhoudsomgeving

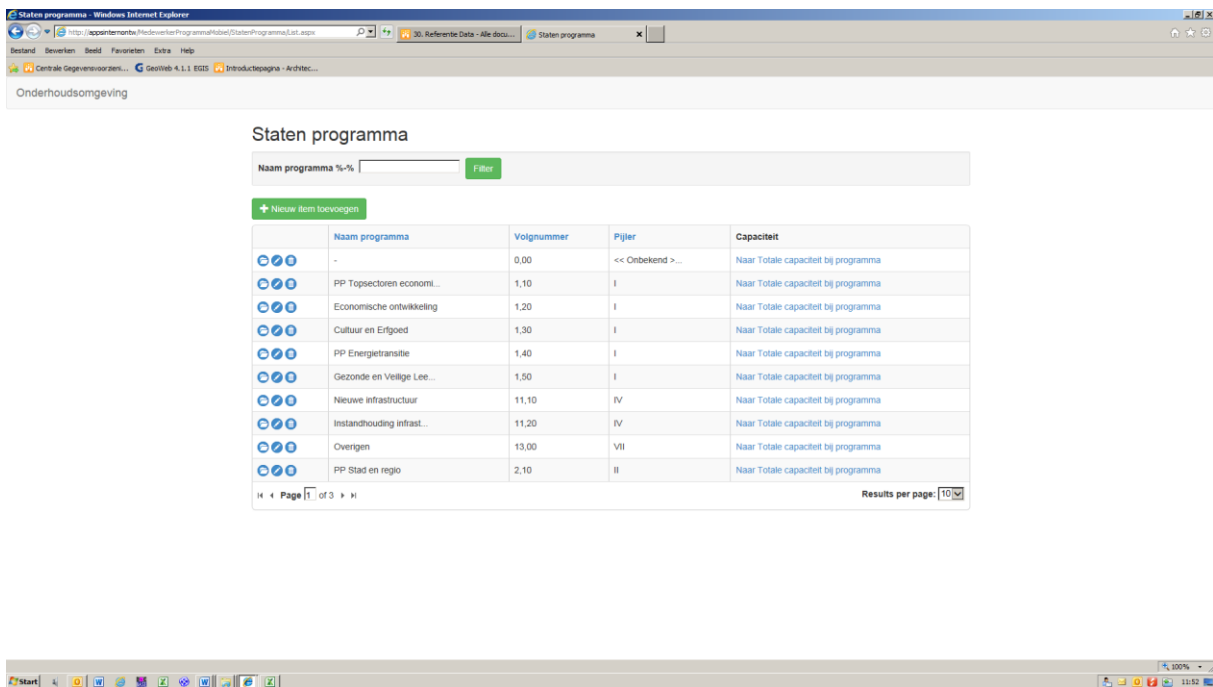
De onderhoudsomgeving kan ingezet worden in een ontwikkeltraject bij de volgende situaties

Gebruikerspopulatie	Ja/Nee
Alleen interne gebruikers	
Beperkte gebruikersgroep	
Gebruikersgroep heeft een redelijke automatiseringskennis	
Functionele requirements	
Gebaseerd op een eenvoudig datamodel	
Eenvoudige invoervalidaties zoals verplicht, datumformaat of keuzelijsten	
Eenvoudige filtering zoals verwijzende sleutels, bevat, hoog-laag etc. Er is in de omgeving een aantal filter opties aanwezig die ingezet kunnen worden (zie de bijbehorende ontwikkeldocumentatie)	
Geen workflow ondersteuning nodig	
Maatwerk	
Alleen configureren van data entiteiten met bestaande datadefinities	
Geen maatwerk programmatuur	
Alleen configureren/ontwikkelen van gedefinieerde extra elementen zoals rapporten, autorisaties of WCF webservices	
Alleen gebruik van de benoemde componenten zoals SQL server reports en WCF	

In alle andere situaties wordt gekozen voor aanschaf van pakketsoftware of een maatwerkontwikkeltraject.

Voorbeeld implementatie

Op basis van de score matrix is een voorbeeld onderhoudsomgeving ingericht op basis van Dynamic Data en het Entity Framework. In onderstaande afbeelding een voorbeeld van de gebruikersinterface. Op de website is daarnaast een template te vinden van een ingerichte omgeving die ingezet kan worden voor een onderhoudsomgeving.



Bijlage scorematrix tooling

Onderdeel	Requirement	Waardering	Dynamic Data	Lightswitch	Score DD	Score LS
Datamodellering						
	Mogelijkheid om een bestaand datamodel uit een database te importeren	5	1	1	5	5
	Diagram weergave voor overzicht van entiteiten	1	2	1	2	1
	Gebruik van business data typen	2	0	1	0	2
	Iteratief ontwikkelen van een datamodel	4	1	1	4	4
	Gebruik van views, tabellen en stored procedures	1	1	1	1	1
	Ontsluiten van webservices bijvoorbeeld Odata zonder extra ontwikkelingsinspanning	2	0	1	0	2
	Mogelijk voor aanvullende modellering zoals captions maar ook domeinen van attributen	2	0	1	0	2
	Integratie met andere ontwikkelparadigma's zoals MVC etc	1	2	1	2	1
	Ontsluiten van views en readonly elementen	3	2	1	6	3
	Vrijheid in definiëren van key constraints	2	2	1	4	2
Schermen						
	Schermen interpreteren op basis van domeinmodel	3	1	0	3	0
	Schermen genereren op basis van domeinmodel	1	0	1	0	1
	Verschillende schermstijlen als template aanwezig zoals lijst detail en combinaties hiervan	2	2	2	4	4
	Master Detail schermen voor ouder kind relaties	2	0	1	0	2

Rijke functionaliteit zonder programmeren	2	1	2	2	4
Mogelijkheid voor calculated columns en properties	1	1	2	1	2
Exportmogelijkheden naar bijvoorbeeld excell	1	0	1	0	1
Inzet van een gelderse huisstijl	1	2	1	2	1
Integratie met andere producten op basis van frames en hyperlinks (van en naar)	3	2	1	6	3
Inzet van views in het datamodel voor generieke datasets	5	1	0	5	0
				0	0
Overigen					
Autorisatie obv SSO binnen toepassing	1	1	2	1	2
Silverlight en HTML versie	1	0	1	0	1
Cloud ready (azure)	1	1	1	1	1
OTAP voor HTML client	1	1	1	1	1
OTAP voor database server	1	1	1	1	1
				51	47