

Automatiseren deployment levert toegevoegde waarde

De Business Intelligence Installer

Jorrit Nijenhuis, Richard Puijk en Vincent Wylenzek

Promoties binnen de OTAP-straat (Ontwikkel-, test-, acceptatie- en productieomgeving) zijn gemeengoed voor elke BI-professional. Zonder promotie is het in principe niet mogelijk om een omgeving 'live' te brengen.

Initiële inrichting en promotie van een omgeving zijn triviaal, maar bij incrementele promoties neemt de kans op fouten door de vele handmatige acties toe. Automatiseren van deployment binnen de OTAP-straat verhoogt de consistentie, kwaliteit, flexibiliteit en snelheid van promoties.

De BI-installer is een hulpmiddel dat deze automatisering mogelijk maakt en daardoor toegevoegde waarde levert voor zowel initiële, incrementele als hotfix promoties. Als vuistregel geldt dat de investering in een BI-installer voor een BI-project met een gemiddelde omvang zich terug verdient binnen enkele volledige promotiecycli. Dit artikel beschrijft eerst het BI-deployment proces, een algemene aanpak hoe initiële en incrementele promoties grotendeels geautomatiseerd kunnen worden, met als doel de foutkans en risico's reduceren. Vervolgens wordt het fenomeen BI-installer behandeld.

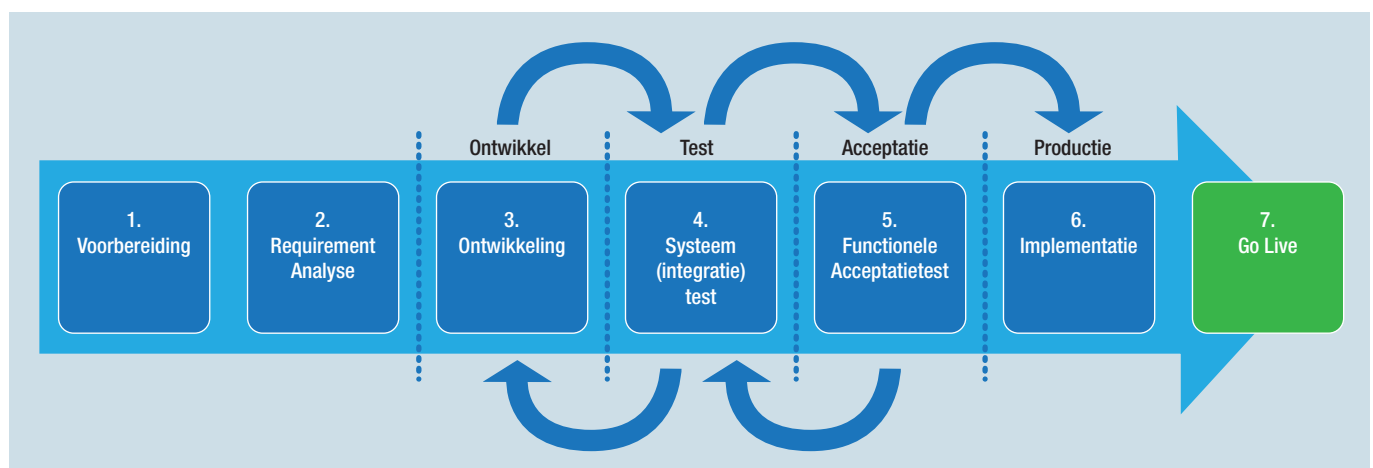
BI deployment (lifecycle) proces

Het werken via de OTAP-straat is populair geworden sinds de introductie van ITIL [1] eind jaren negentig. Damhof [2] stelt dat deze methode voor huidige begrippen te traag is en stelt een meer opportunistische systeemontwikkeling voor.

Opportunistisch impliceert korte ontwikkelcycli en snel kunnen

reageren op de klantvraag. Ook Harland [3] stelt de behoefte voor OTAP met betrekking tot BI ter discussie. Kort samengevat, het wel of niet gebruik maken van OTAP is op dit moment een interessante discussie. Wij denken dat door het toepassen van de BI-installer het OTAP-proces aanzienlijk versneld kan worden, waardoor de traditionele systematische aanpak weer dichterbij de opportunistische aanpak aan komt te liggen. Op dit moment zijn de projecten waarbij wel een (gedeeltelijke) OTAP-straat aanwezig is nog steeds in de meerderheid. Wij verwachten dat dit op korte termijn niet wijzigt, wat voldoende beweegredenen biedt om het deployment proces te automatiseren.

De beschreven deploymentcyclus richt zich dan ook op OTAP, namelijk het proces van het ontwikkelen van software op een ontwikkelomgeving tot en met de finale uitrol op de productieomgeving. Afbeelding 1 brengt de ontwikkelstappen in relatie met de deploymentfases in kaart. Zoals de pijlen aangeven zijn



Afbeelding 1: Deploymentcyclus.

promoties en demoties mogelijk. Onder promotie wordt een verhoging van het server level verstaan, dit is bijvoorbeeld van ontwikkelfase naar testfase. Onder een demotie wordt een verlaging van het server level verstaan, dit is bijvoorbeeld van productiefase terug naar ontwikkelfase.

Deployment

Bij een initiële promotie van een omgeving maken wij het onderscheid tussen de installatie van respectievelijk hardware, software en BI-programmatuur. Tijdens een incrementele promotie is alleen de laatstgenoemde categorie relevant, want de rest is al ingericht. De hardware- en software-installatie vallen daarom buiten de scope van dit artikel. We bespreken achtereenvolgens de kenmerken van een initiële en incrementele promotie.

Initiële promotie. De eerste keer dat de BI-programmatuur op een omgeving wordt geïnstalleerd (de initiële promotie) is relatief eenvoudig. Alle databaseobjecten kunnen worden aangemaakt en ETL-logica, OLAP-cubes, rapporten en het management dashboard kunnen worden uitgerold.

Incrementele promotie. Indien de BI-programmatuur eenmaal in productie is genomen, is er geen initiële promotie meer mogelijk. De tabellen zijn dan al gevuld en kunnen niet zomaar opnieuw worden aangemaakt. Toch moet ook de productiesituatie gewijzigd kunnen worden. Enerzijds kunnen er nieuwe ontwikkelingen zijn en anderzijds kunnen er nog fouten aan het licht komen tijdens de beheerfase. Bij een promotie met als doel het verbeteren van deze fouten spreekt men van een hotfix.

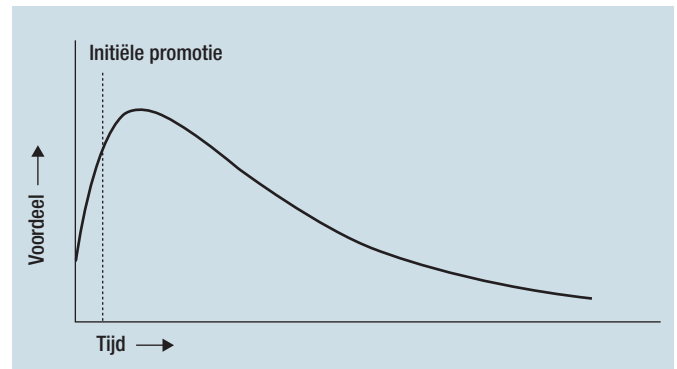
Deploymentproblematiek. Helaas impliceert een promotie een groot aantal stappen. Dit vergt op de traditionele manier het nodige aan discipline en documentatie. In de praktijk loopt men vaak tegen terugkerende problemen aan. Hoe vaak komt het niet voor dat de omgevingsvariabelen niet worden aangepast, waardoor de verkeerde omgeving wordt geladen, of erger; wordt geleegd? Door het promotieproces grotendeels te automatiseren kunnen we dit risico minimaliseren.

De behoefte aan het automatiseren van het promotieproces is het grootst vlak na de initiële promotie. Er zullen dan waarschijnlijk nog relatief veel hotfixes nodig zijn. Wanneer de frequentie van de hotfixes na verloop van de tijd afneemt, is een geautomatiseerd deploymentproces van minder groot belang. Bovenstaande logica is tevens terug te vinden in afbeelding 2.

Wanneer meerdere ontwikkelingen zich in verschillende fases van de OTAP-straat bevinden, neemt de complexiteit van versiebeheer en release management toe. De BI-installer kan deze problematiek eenvoudiger maken, doordat de verschillende ontwikkelingen eenvoudig en snel na elkaar gepromoot kunnen worden.

De BI-installer

De BI-installer (zie afbeelding 3) is de praktische uitwerking van het idee voor een automatische uitrol. Het initiatief voor dit hulp-



Afbeelding 2: Voordeel versus tijd grafiek.

middel is ontstaan vanuit de eerder vermelde problematiek. De BI-installer is ontwikkeld in C# voor het Microsoft platform en is momenteel ingericht voor SQL Server 2008.

De BI-installer is te gebruiken voor het deployen van fysieke bestanden, DDL- en DML-scripts, datascripts, ETL-packages (SSIS), OLAP-cubes (SSAS) en Rapporten (SSRS). Voor al deze componenten wordt zoveel mogelijk gebruik gemaakt van de standaardfunctionaliteit van SQL Server 2008 en bijbehorende tools (SSIS, SSAS en SSRS). Elk onderdeel kent zijn eigen verwerkingsmechanisme, hieronder toegelicht.

Fysieke bestanden. De class System.IO.File wordt gebruikt om bestanden uit een bronfolder te kopiëren naar een doelfolder. De locatie van deze folder is configureerbaar door middel van de user interface van de BI-installer.

Databasescripts (DDL, DML en data). Er zijn diverse tools voorhanden waarmee databasescripts kunnen worden gegenereerd. Wij maken gebruik van Visual Studio 2008 – Database Edition GDR (R2). GDR (R2) is een add-in welke ook bekend is onder de naam 'Data Dude'. De exacte werking van deze tool valt buiten de scope van dit artikel en is eventueel na te lezen bij Drapers [4]. Met behulp van deze tool kunnen we per database initiële scripts maar ook deltascripts genereren. Omdat data, los van omgevingsafhankelijke variabelen, niet identiek overgenomen dienen te worden naar elke omgeving, maken wij gebruik van een stored procedure met een roll-back mechanisme. Deze procedure wordt aangeroepen door de BI-installer. Binnen Visual Studio worden alle componenten voortdurend gevalideerd. Kritieke fouten moeten opgelost worden voordat de scripts gegenereerd kunnen worden. De scripts worden door middel van een SQLCommand commando aangeroepen (voorbeeld syntax: `Sqlcmd -s <Servername> -v <Omgevingsvariabele[1]> -v <Omgevingsvariabele[n]>`).

ETL-Packages. SSIS-packages kunnen fysiek in de MSDB-database of op het bestandssysteem worden opgeslagen. Er zijn op het web talloze discussies te vinden over de voor- en nadelen van elk van deze twee methoden. Wij maken gebruik van de MSDB-database.

Deployment van deze packages kan op diverse manieren gebeuren. De BI-installer maakt gebruik van de method LoadPackage

van de Class Microsoft.SqlServer.Dts.Runtime.Application. Deze methode is erg snel en de voortgang van de deployment wordt geretourneerd naar het scherm. In geval van opslag in het bestandssysteem kan gebruik gemaakt worden van de beschreven methode voor fysieke bestanden.

OLAP-Cubes. Voor het deployen van de SQL Server Analysis Services OLAP-Cubes maken wij gebruik van de Microsoft.AnalysisServices.AdomdClient-class. Deze voert de XMLa-scripts uit de meegegeven mappenstructuur uit. De XMLa-scripts worden gegenereerd op basis van de .asdatabase file. Deze file wordt weer gegenereerd door Visual Studio bij elke willekeurige deployment. Dit .asdatabase script is niet één op één om te zetten in XMLa, vanwege niet te interpreteren auditinformatie. Deze auditinformatie is te herkennen aan XML-tags en is daarmee eenvoudig geautomatiseerd te verwijderen.

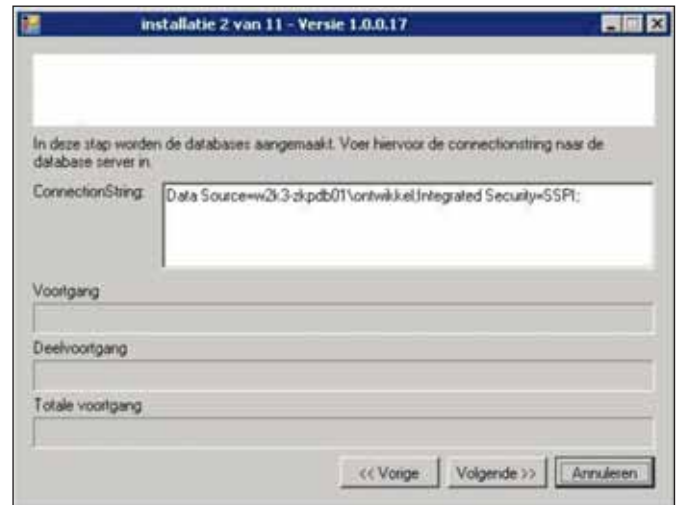
Rapporten. De reportservice2006.asmx service maakt het mogelijk om rapporten te deployen naar een SQL Server Report Server welke geïntegreerd is met SharePoint. Hierbij wordt de locatie van de datasource via een variabele door de BI-Installer meegegeven. Bij een 'stand-alone report server' draaiende onder SQL Server 2005 of 2008 r1, zal gebruik gemaakt moeten worden van de reportservice2005.asmx service. SQL Server 2008 r2 maakt gebruik van een nieuwe versie van deze reportservice met de naam: reportservice2010.asmx.

Ontwikkelen BI-installer

Scope. Alvorens de BI-installer gerealiseerd kan worden, is het van belang het toepassingsgebied te bepalen. Het toepassingsgebied kan zijn een initiële installatie, incrementele installatie of bijvoorbeeld hotfixes. Voordat we het deploymentproces kunnen automatiseren, moeten we duidelijkheid hebben over welke componenten onderdeel zijn van de promotie. Dit kunnen scripts zijn die uitgevoerd moeten worden, rapporten die gedeployed dienen te worden en DTSX-packages die in de MSDB of het filesystem moeten worden geplaatst, maar ook zaken zoals een 'web .config', analysis services cubes en eventuele hulpapplicaties moeten niet worden vergeten.

Mappenstructuur. Om de BI-installer multi-toepasbaar te maken is het van belang een gescheiden mappenstructuur te gebruiken. Hierdoor wordt het eenvoudig om per component te achterhalen waar de installatiebestanden zich bevinden. In de programmacode kan worden bepaald wat er met de bestanden in de desbetreffende map kan worden gedaan. Zoals kopiëren uitvoeren, aanpassen enzovoort. Door de naamgeving van bestanden en het gebruik van submappen wordt het mogelijk de volgorde van uitvoer te beïnvloeden. Door mappen weg te laten, kunnen er ook delen uit de BI-installer worden gehaald. Dit is handig bij het uitrollen van hotfixes. De tabel in afbeelding 4 toont een voorbeeldopzet voor een mappenstructuur.

Omgevingsvariabelen. Vervolgens dient nagedacht te worden over een zo generiek mogelijke oplossing. Op deze manier wordt de BI-installer multi-inzetbaar en kan worden ingezet voor initiële installatie, incrementele installatie, promoties, demoties, hot-



Afbeelding 3: Screenshot Business Intelligence Installer.

fixes enzovoort. Om de foutkans te minimaliseren kan men gebruik maken van een configuratiebestand, met daarin de omgevingsafhankelijke variabelen. Een dergelijke configuratie wordt eenmalig ingericht per omgeving en voorkomt dat variabelen naar verkeerde omgevingen verwijzen.

Foutafhandeling. Verder is het van essentieel belang om na te denken over back-up en roll-back mechanismen. Het is bijvoorbeeld mogelijk om een dergelijk mechanisme te implementeren in de databasescripts, maar het kan ook als een separate stap worden opgenomen in de uitrolprocedure. Datascripts moeten in ieder geval voorzien worden van een roll-back mogelijkheid, zodat de reeds toegevoegde, bewerkte of verwijderde records uit een foutief script, in hun oorspronkelijke situatie worden hersteld. Voor alle overige componenten kan men de foutmeldingen verkrijgen door de error events van de betreffende methods af te vangen.

Testen. Tot slot zullen we dit uitgebreid moeten testen. Want als het bij een geautomatiseerd proces fout gaat, werkt dit door in alle gerelateerde omgevingen.

Inrichten BI-installer

Na het ontwikkelen van de BI-installer zal deze bij een specifieke promotie ook moeten worden ingericht.

Mappenstructuur. De mappenstructuur, zoals deze hiervoor is besproken, zal gevuld worden met de diverse componenten. Voor de databaseobjecten bevat de map de delta's tussen de bron en doelomgeving. Voor andere componenten zoals bijvoorbeeld de packages, geldt dat juist de nieuwste situatie wordt neergezet. Het is dan niet nodig om de delta's te bepalen.

Release notes. Tot slot is het belangrijk om release notes op te nemen. Dit kan in diverse vormen, een plat tekstbestand voldoet prima. In de release notes staan de acties beschreven welke door de consultant ante- of post-deployment moeten worden uitgevoerd. Een ante-actie is bijvoorbeeld het controleren op unieke business keys binnen een tabel, zodat de unique key constraint uit een promotieactie succesvol kan worden doorgevoerd.

Map	Toepassingsgebied
SQLScripts\Initieel	De DDL en DML scripts voor een initiële installatie
SQLscripts\Incrementeel	De DDL en DML scripts voor een incrementele installatie
DtsxSource	De DTSX packages
FileSource	Extra bestanden voor eventuele hulpapplicaties
RdlSource	De te deployen rapporten
SsasSource	De XMLA kubusscripts
Log	De logfiles van de BI-Installer

Afbeelding 4: Mappenstructuur.

Voordelen van de BI-installer

Het gebruik van een BI-installer biedt een aantal voordelen.

Kwaliteit en consistentie. Door de BI-installer te gebruiken, zal een hotfix exact hetzelfde worden uitgevoerd op iedere omgeving. Dit borgt de kwaliteit en consistentie van de oplossing.

Flexibiliteit. De BI-installer biedt grote flexibiliteit. Door de generieke opzet is de BI-installer multi-toepasbaar voor onder andere promoties, initiële en incrementele installaties.

Overdraagbaar. In veel organisaties is een beheerorganisatie verantwoordelijk voor de productieomgeving en is het niet mogelijk om hier als supplier zelf rechtstreeks wijzigingen op uit te voeren. Door de BI-installer is een upgrade of hotfix eenvoudig overdraagbaar aan een derde partij.

Snelheid. Een promotie kan sneller worden uitgevoerd door de BI-installer, waarin veel stappen zijn geautomatiseerd. De BI-installer werkt met een initialisatiebestand, waarin alle omgevingsvariabelen eenmalig worden opgegeven.

Nadelen

Naast voordelen zijn er natuurlijk ook enkele nadelen te noemen bij het gebruik van de BI-installer.

Andere manier van werken. De ontwikkelaars dienen volgens een vooraf bepaalde structuur te werken. Op deze manier kan worden geborgd dat de omgevingen binnen de OTAP-straat identiek blijven. Dit betekent consistentie in de uitvoering van het werk en moet voor professionals geen probleem vormen.

Automatiseren aan het begin van het project. Het nadeel van een BI-installer is dat er bij het begin van het project, wanneer er al genoeg andere aandachtspunten zijn, energie gestoken moet worden in het automatiseren van de diverse stappen. Door de BI-installer uit te stellen onder druk van budget en deadlines, nemen de voordelen (juist in het begin van het beheertraject) af. Zie ook afbeelding 2.

Een dergelijke investering verdient zich echter snel terug. Het break-even moment hangt sterk af van het aantal en de omvang van de promoties. Voor een gemiddeld project is het break-even punt al na enkele volledige cycli bereikt.

Kwaliteit en impact. Indien de scripts binnen de BI-installer fouten bevatten en dit niet opgelost is voor de uitrol, heeft dit invloed op alle door de BI-installer geraakte omgevingen. Als preventieve maatregel moet men goed nadenken over het debug- en testproces.

Samenvatting

Het automatiseren van deployment binnen de OTAP-straat verhoogt de consistentie, kwaliteit, flexibiliteit en snelheid van promoties. In het algemeen mogen we concluderen dat een BI-installer toegevoegde waarde levert voor zowel initiële, incrementele en hotfix promoties. Het is echter essentieel om de BI-installer zorgvuldig in te richten en te testen, alvorens deze in te zetten.

Voor een Microsoft BI-project kan redelijk eenvoudig een BI-installer worden gecreëerd. Voor (combinaties van) andere tools biedt het artikel geen technische uitwerking, maar de gehanteerde stappen kunnen natuurlijk ook hier als uitgangspunt worden genomen.

Als vuistregel geldt dat de investering in een BI-installer voor een BI-project met een gemiddelde omvang zich terug verdient binnen enkele volledige promotiecycli. Uiteraard moet bij de start van het project wel voldoende ruimte worden vrijgemaakt in de vorm van doorlooptijd en geld.

Literatuur

- [1] Bon, Jan van (2008). *IT-Service Management op basis van ITIL V3. Eerste druk, Van Haren Publishing, Zaltbommel.*
- [2] Damhof, Ronald (2009). *Development Processes in Data Warehouse Environments.*
www.b-eye-network.co.uk/channels/1551/view/10744/
- [3] Harland, Frank (2009). *OTAP en Datawarehouse.*
<http://franklybi.blogspot.com/2009/11/otap-en-datawarehouse.html>
- [4] Drapers, Gert (2010). *Home of the Data Dude.*
<http://blogs.msdn.com/b/gertd/>

Jorrit Nijenhuis is Business Intelligence Consultant bij Ordina.

Richard Puijk is Business Intelligence Consultant bij Ordina.

Vincent Wylenzek is Senior Business Intelligence Consultant bij Ordina.