

Praktijkervaringen met consistent datawarehouse

De waarde van architectuur voor beheer

Henk Binnendijk

Kenmerkend voor een DWH/BI-project is dat het dynamisch is en dat er na oplevering regelmatig om wijzigingen gevraagd wordt. Deze wijzigingen komen voort vanuit de business of vanuit de bron; de eerste via changes en/of requests en de tweede vaak door incidenten. In dit artikel worden ervaringen gedeeld over het continu beheren van dynamische DWH/BI-omgevingen. Hoe voorkom je dat je een inflexibele of niet-continue BI-omgeving hebt?

Door de wisselende informatiebehoefte van de business is het wenselijk om een productionele BI-omgeving continu te kunnen aanpassen. Vanuit beheersperspectief geldt juist het tegenovergestelde. Hoe minder veranderingen nodig zijn des te stabiel is de omgeving, des te hoger is de beschikbaarheid en des lager zijn de kosten. Bij DWH/BI-projecten is het een bekend fenomeen dat de hoeveelheid veranderingen vele malen hoger ligt dan bij andere automatiseringsprojecten. Nieuwbouw BI-projecten worden om deze reden door middel van incrementen (bijvoorbeeld volgens het Kimball-principe) opgeleverd in plaats van met een *big bang* (zoals wel bekend is van Inmon). Zelfs al ligt de frequentie van doorvoer van releases hoog (bijvoorbeeld een keer per maand), dan nog zijn de business requirements vaak alweer veranderd. In het beheer van BI-omgevingen wordt gesproken over changes in plaats van releases. Changes worden na goedkeuring direct in productie genomen. Dit komt de business ten goede, omdat er beter gestuurd kan worden in dynamische marktomstandigheden. De stabiliteit van de productieomgeving staat hierdoor echter wel onder druk. Het beheer van een DWH/BI-omgeving met hoge wijzigingsfrequentie kan het beste ondersteund worden via (minimaal) een OAP-straat. Daarnaast is het belangrijk dat ontwikkelaars en beheerders nauw samenwerken. Het is het beste als ontwikkelaars en beheerders in een team zitten.

Er wordt nog wel eens gedacht: "het is minder erg als er zich fouten voordoen in de rapportageomgeving, dan in ons orderproces". Deze gedachte is meestal niet juist. Het is niet altijd direct zichtbaar waar de fout zit, helemaal niet als het om een functionele fout gaat, die er voor zorgt dat er foute cijfers gerapporteerd worden. Maar hierdoor kan een managementbeslissing ernstige gevolgen hebben. Als een artikel met het karakter 'aantal' aangeleverd is met het karakter 'volume', dan

gaat de berekening technisch goed. De functionele uitkomst kan echter wel zijn dat de omzet ineens tien keer hoger uitvalt voor dit artikel. Met als gevolg een inkoopactie om in de voorspelde toekomstige omzet te kunnen voorzien. Hierdoor gaat de voorraadwaarde omhoog. En zo ontstaat er een sneeuwbaaleffect dat veroorzaakt wordt door een kleine fout in de management-informatievoorziening.

Architectuur

Goede (bijvoorbeeld ITIL-gedreven) beheerprocessen alleen zijn niet voldoende om een BI-omgeving beheersbaar *en* flexibel te houden. ITIL is een referentiekader om de ICT (beheer) processen binnen een organisatie in te richten. Over (datawarehouse) inhoudelijke vraagstukken zegt ITIL niets. In het proces dient inhoudelijke Business Intelligence- en datawarehouse-kennis 'gestopt' te worden om issues en vraagstukken van de organisatie snel op te lossen.

Een gedegen technische architectuur is benodigd om binnen beheerprocessen zowel flexibiliteit als continuïteit te bieden. Het is gebruikelijk om een impact-analyse te maken van een gewenste wijziging op een bestaande BI-omgeving. Hiervoor is vaak weinig tijd. Daarbij komt nog dat degene die de impact-analyse moet maken vaak niet dezelfde persoon is als degene die de omgeving gebouwd heeft. Goede architectuurdocumentatie zorgt ervoor dat impact-analyses veel sneller en beter gemaakt kunnen worden.

ITIL als beheerproces en datawarehouse-architectuur als fundament voor de management-informatievoorziening borgen kennis en continuïteit in het beheer en aanpassingen van systemen. Een voorbeeld hiervan is het vastleggen van gegevensstromen in een dataflow-diagram. Zonder een dataflow-diagram dient eerst de hele gegevensstroom onderzocht te worden als er een issue

optreedt of een wijziging doorgevoerd moet worden. Het diagram biedt inzicht en daarmee de mogelijkheid om gericht te analyseren. Architectuur voorkomt willekeur in de keuze van oplossingen en draagt bij aan consistentie van een datawarehouse. Het voorkomt onnodig zoekwerk en versnelt daarmee het beheerproces en het in productie nemen van aanpassingen. Bovendien maken duidelijke architectuurkeuzes, die ook gedocumenteerd zijn, de ontwikkeling van een datawarehouse minder persoonsgebonden.

In het algemeen zorgt architectuur voor eenduidige kaders waarbinnen een datawarehouse ontworpen kan worden. ITIL zorgt samen met architectuur voor de continuïteit en flexibiliteit. Changes en incidenten kunnen snel opgepakt worden en worden niet vertraagd door bijvoorbeeld het vertrek van een kennishebbende medewerker. Tevens draagt architectuur bij aan een eenduidigheid van het totale IT-landschap.

Architectuurdocumentatie

Incidenten worden vaak opgelost met een work-around, omdat de feitelijke oorzaak niet ontdekt wordt. Architectuurdocumentatie is zeer belangrijk bij het snel op de juiste wijze oplossen van een incident. Bij het oplossen van een incident gaat het grootste deel van de tijd zitten in het onderzoek naar het probleem. De oorzaak van het incident verhelpen is vervolgens snel gebeurd. Een goed architectuurplaatje is vaak voldoende om het onderzoek naar de oorzaak van het probleem te lokaliseren op de juiste plek. Als de echte oorzaak niet bekend is, is het niet mogelijk om een structurele oplossing door te voeren. Het komt dikwijls voor dat een beheerder door een foutmelding in eerste instantie op het verkeerde been wordt gezet. Een foutmelding kan namelijk verwijzen naar een vervolgfout in plaats van naar de feitelijke oorzaak van het probleem. Goede architectuurdocumentatie zorgt ervoor dat dit niet gebeurt. Is deze documentatie er wel, dan ben je feitelijk met een change bezig. Als bijvoorbeeld een artikel twee keer voorkomt in de dimensie kan de oorzaak zich feitelijk in de staging area bevinden.

Changes worden na goedkeuring direct in productie genomen

Het verzorgen van architectuurdocumentatie hoeft niet veel tijd te kosten. Het is noodzakelijk om naar een juiste balans te zoeken in wat men wel en niet documenteert. Te veel documentatie zorgt ervoor dat onderhoud van documentatie veel tijd en geld kost. Het gevaar is dan dat het onderhoud niet of niet goed gebeurt. Gevolg is dat de documentatie niet meer vertrouwd wordt en dat deze dus nutteloos is. Het compleet documenteren van bijvoorbeeld een ETL dataflow/mapping heeft geen zin en kost veel tijd. De dataflow vertelt zelf wat het technisch doet. Het is veel belangrijker om kort te beschrijven wat het doel van

Documentatie

Goede architectuurdocumentatie voor beheer bevat minimaal:

- Overall architectuuroverzicht (lagenmodel);
- Dataflow-diagram;
- Datamodel(len);
- Hiërarchie-overzichten;
- Dataverwerkingstechniek (incrementeel, datacorrectiemogelijkheden);
- Standaarden (bijvoorbeeld naamgeving) en richtlijnen (wel of geen views gebruiken);
- Doorlooptijdschema;
- Hardware-overzicht;
- Randvoorwaarden overzicht (backup-procedure, gebruikerswindows, enzovoort).

Bij het documenteren is het met name belangrijk om de 'waarom' vraag vast te leggen.

de techniek is en onder welke randvoorwaarden het is gemaakt. Als voorbeeld: in de dataverwerking wordt organisatie nummer 10 gefilterd. De dataflow vertelt dat dit gebeurt, maar niet waarom (het is een dummy organisatie-onderdeel).

Gebrek aan architectuur

Gartner heeft uitgezocht dat de kosten van een BI-programma voor 30 procent bestaan uit de initiële ontwikkelingskosten en voor 70 procent uit beheerkosten. Bij beheer wordt dan bedoeld: het in stand houden en doorontwikkelen van de BI-applicatie op initieel ontwikkelde functionele gebieden. Maar hoe komt het dat er zoveel projecten zijn waar een goede (met name architectuur-) documentatie ontbreekt? Of erger: waar al helemaal een architectuur (visie) ontbreekt. Dit wordt vaak veroorzaakt door twee fenomenen.

1. Managers die investeringsbeslissingen nemen voor een BI-project hebben te maken met budgetten en de te halen targets. De termijn waarbinnen de targets gehaald moeten worden (meestal een fiscaal of boekhoudkundig of kalenderjaar) is vaak korter dan de lifecycle van een BI-project. Tijdens de ontwikkeling van een BI-project heeft men te kampen met tegenslagen. Tegenslagen die vooraf niet in tijd of geld begroot zijn. Vervolgens worden er aanpassingen gedaan om de gestelde projectdoelstellingen toch te halen. Documentatie wordt vaak als eerste niet meer als noodzakelijk gezien en dus geschrapt. Of er wordt druk uitgeoefend op de ontwikkelaar om z'n werk sneller op te leveren waardoor de ontwikkelaar gaat afwijken van de architectuur. Aangenomen uiteraard dat er al helemaal is stilgestaan bij een goede architectuur. Te vaak wordt gedacht dat architectuur niet nodig is en wordt het geschrapt om kosten te besparen.
2. Er is een architectuur opgezet en zelfs gedocumenteerd, maar

er wordt nagelaten om deze te beheren. Dat komt onder meer doordat vaak de rol van architect alleen bij het begin van een project aanwezig is. De architect is op dat moment betrokken bij het bepalen van de functionele en technische architectuur. Soms helpt hij mee in de keuze van welke BI-software en hardware nodig is. Zodra de bouwfase start gaat de architect weg. Men lijkt hierbij de rol van architect te kopiëren vanuit de bouwwereld. Echter bij BI-projecten is het van wezenlijk belang dat na het bepalen van een architectuur deze ook tijdens de ontwikkeling en beheer getoetst wordt. De toetsing dient auditair te zijn

Compleet documenteren van een ETL dataflow heeft geen zin en kost veel tijd

op technisch vlak. Op functioneel vlak dient de architectuur steeds getoetst te worden ten aanzien van de almaar wijzigende informatiebehoefte van het management. Hoe eerder dit gebeurt, hoe 'schoner' het project wordt opgeleverd. Hoe eerder fouten worden ontdekt, hoe meer schaalbaar en flexibel de BI-omgeving in de toekomst is. En nog belangrijker; hoe beheerbaarder en dus goedkoper op de langere termijn.

Het bewaken van de architectuur is net zo belangrijk tijdens de beheerfase! Het is feitelijk een van de belangrijkste middelen om een toekomstvaste BI-omgeving te kunnen (blijven) garanderen.

Conclusie

Er kan geconcludeerd worden dat het voor BI-projecten beter is dat de gestelde management budgetten en targets gelijk zijn aan de complete lifecycle van een omgeving in plaats van de veelal voorkomende fiscale jaargangen. Uiteraard zijn er nog veel meer efficiency-punten aan te dragen om de beheerkosten te minimaliseren, maar de grootste besparing op beheerkosten wordt niet in de beheerfase zelf gemaakt, maar bij het ontwikkelen tijdens het project. Belangrijk daarbij is de rol van de architect gedurende de gehele lifecycle van een BI/DWH-programma.

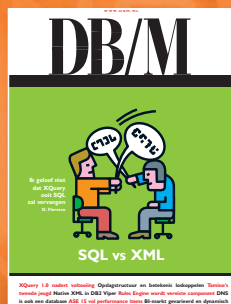
In een ITIL-gedreven organisatie is architectuur een noodzakelijke aanvulling om doorontwikkelingen snel te kunnen opleveren. Architectuur biedt de inhoudelijke aanvulling op het procesgedreven ITIL. Met een eenduidige, bewust gekozen, architectuur is men ook beter in staat om incidenten snel op te lossen.

Om een project beheersbaar maar ook flexibel te houden is het belangrijk dat er een goede architectuurvisie is en dat deze gedocumenteerd is (niet te gedetailleerd). Het belangrijkste van architectuur, voor een goede continuïteit en flexibiliteit, is dat het bewaakt en onderhouden wordt.

Henk Binnendijk is Business Intelligence/Datawarehouse Consultant bij i3.

Neem nu een abonnement op

Database Magazine



Méér dan een vakblad!

www.dbm.nl

Database Magazine is al sinds 1990 een **kwaliteitsvakblad** op het gebied van databeheer, databases, datawarehousing en Business Intelligence. Onderwerpen als relationele en objectgeoriënteerde DBMS-en, datamodellering, database-ontwerp, datakwaliteit, datawarehousing, datamanagement, metadata, tools, datamigratie, data-integratie, performance, OLAP, ETL, Web Services, Business Intelligence, Analytic Applications, SQL, XML... u komt ze acht maal per jaar tegen in DB/M. Met vaak praktische tips en voorbeelden. De praktijk staat ook voorop in zo'n 30 **seminars**, masterclasses en congressen die jaarlijks speciaal voor u worden georganiseerd.

Database Magazine heeft een uitgebreide website met onder andere:

- actueel nieuws,
- agenda met events,
- compleet overzicht van alle producten in de markt,
- een database met detail-informatie van BI-tools: de BI-Matrix,
- het online archief met alle artikelen uit het blad, te downloaden door abonnees,
- een overzicht van relevante vacatures op uw vakgebied.

Ook kunt u zich kosteloos abonneren op de **e-mail nieuwsbrief** die eenmaal per 3 weken verschijnt en geheel in het teken staat van Business Intelligence en Datawarehousing.

Nog geen abonnee?

Meld u online aan op www.dbm.nl. Het eerste jaar profiteert u van bijna 50% korting voor nieuwe abonnees.