



De waarde van architectuur voor beheer

Enkele praktijk ervaringen

Door Henk Binnendijk

Kenmerkend aan een DWH / BI project is dat het dynamisch is en dat er na oplevering regelmatig om wijzigingen gevraagd wordt. De 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 inflexibel en/of niet continue BI omgeving hebt?

Beheer

Door de steeds wisselde informatiebehoefte van de business is het wenselijk om een productionele BI omgeving continu te kunnen aanpassen. Vanuit het perspectief van beheer 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 velen malen hoger ligt dan bij andere automatiseringsprojecten. Nieuwbouw BI projecten worden om deze reden middels incrementen (b.v. 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 1 per maand), dan nog zijn de business requirements vaak alweer veranderd. In 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 productie omgeving 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.

In bedrijven wordt nog wel eens gedacht: "Het is minder erg als er zich fouten voordoen in de rapportage omgeving, 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 management beslissing 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 10 keer hoger uitvalt voor dit artikel. Met als gevolg een inkoop actie om in de voorspelde toekomstige omzet te kunnen voorzien.

Architectuur

Goede (bijvoorbeeld ITIL gedreven) beheerprocessen alleen zijn niet voldoende om een BI omgeving beheersbaar EN flexibel te houden. Een gedegen technische architectuur is nodig om binnen de beheerprocessen zowel flexibiliteit als continuïteit te bieden. Het is gebruikelijk een impact analyse te doen van een gewenste wijzigingen 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 architectuur documentatie zorgt ervoor dat impact analyses veel sneller en beter gemaakt kunnen worden.

Architectuur documentatie

Incidenten worden vaak opgelost met een workaround, omdat de feitelijke oorzaak niet ontdekt wordt. Architectuur documentatie is zeer belangrijk om snel een incident op de juiste wijze op te lossen. Bij het oplossen van een incident gaat het grootste deel van de tijd zitten in het onderzoek naar wat het probleem is. De oorzaak van het incident verhelpen is vervolgens snel gebeurd. Een goed architectuur plaatje 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.

Let's do business intelligence



Goede architectuur documentatie zorgt ervoor dat dit niet gebeurt. Is de documentatie er wel, dan ben je feitelijk met een change bezig. Als bijvoorbeeld een artikel 2 keer voorkomt in de dimensie kan de oorzaak zich feitelijk in de staging area bevinden.

Het verzorgen van architectuur documentatie hoeft niet veel tijd te kosten. Het is noodzakelijk om te zoeken naar een juiste balans in wat je wel en niet documenteert. Te veel documentatie zorgt ervoor dat het onderhoud van de documentatie veel tijd en geld kost. Het gevaar is dan dat het onderhoud van de documentatie niet of niet goed gebeurt. Het gevolg is 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 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% bestaat uit de initiële ontwikkeling- en voor 70% 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 goede (met name architectuur) documentatie ontbreekt? Of erger: waar überhaupt een architectuur (visie) ontbreekt. Dit wordt vaak veroorzaakt door de volgende 2 fenomenen.

1. Managers die investeringsbeslissingen nemen voor een BI project hebben te maken met budgetten en te behalen targets. De termijn waarbinnen de targets behaald moeten worden (meestal een fiscaal/boekhoudkundig/kalender jaar) 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 project doelstellingen 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 überhaupt al stil is gestaan bij een goede architectuur. Te vaak wordt gedacht dat architectuur niet nodig is en geschrapt wordt om kosten te besparen.
2. Er is een architectuur gemaakt en zelfs gedocumenteerd, maar er wordt nagelaten om dit 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 / zij 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 op technisch vlak. Op functioneel vlak dient de architectuur steeds getoetst te worden ten aanzien van de alsmaar wijzigende informatiebehoefte van het management. Hoe eerder dit gebeurt hoe "schoner" het project wordt opgeleverd. Hoe eerder fouten worden ontdekt, hoe 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 trouwens net zo belangrijk tijdens de beheer fase! Het is feitelijk een van de belangrijkste middelen om een toekomstvaste BI omgeving te kunnen (blijven) garanderen.

Conclusie

Uit bovenstaand verhaal 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.

Let's do business intelligence



Uiteraard zijn er nog veel meer efficiency punten aan te dragen om de beheerkosten te minimaliseren, maar de grootste besparing op beheerkosten worden niet in de beheerfase zelf gemaakt, maar bij het ontwikkelen tijdens het project. Belangrijk daarbij is de rol van de architect gedurende de gehele life-cycle van een BI/DWH programma.

Om een project beheersbaar maar ook flexibel te houden is het belangrijk dat er een goede architectuur visie 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.

Goede architectuur documentatie voor beheer bevat minimaal

- Overall architectuur overzicht (lagen model)
- Dataflow diagram
- Datamodel(len)
- Hiërarchieoverzichten
- Dataverwerkingstechniek (incrementeel, datacorrectie mogelijkheden)
- Doorlooptijd schema
- Hardware overzicht
- Randvoorwaarden overzicht (back-up procedure, gebruikers Windows, etc.)