

Een standaard ontwikkelmethode bestaat niet

Doe-het-zelf voor BI

Jan-Paul Fillié

Menig Business Intelligence-project wordt uitgevoerd zonder dat een duidelijk beeld bestaat van waar naartoe gewerkt moet worden. Dat is ook de reden dat veel van dergelijke projecten meer kosten of minder functionaliteit bieden dan vooraf ingeschat.

De meeste aanbieders van BI-projecten hanteren eigen methodes zoals de waterval-methode en in een enkel geval enige time-boxing. Het gevolg is dat binnen BI-projecten altijd onderdeel voor onderdeel en van achteren (bronnen) naar voren (bijvoorbeeld rapportage) wordt gewerkt. Dit zorgt voor een slechte spreiding van de risico's.

Grip

In de afgelopen jaren is vaak getracht om standaard software-ontwikkelmethoden toe te passen op BI, maar daarbij is geconstateerd dat BI te veel afwijkt van ontwikkeling van een operationeel systeem. Iteratieve methodes, zoals Dynamic Systems Development Method (DSDM), werken niet omdat de gebruikersinteractie met het systeem vrijwel altijd beperkt is tot een klein deel van het systeem of een gegeven is door de keuze van een

tool. Hierdoor wordt het complete datawarehouse beschouwd als een vereiste voor één OLAP-kubus of rapport.

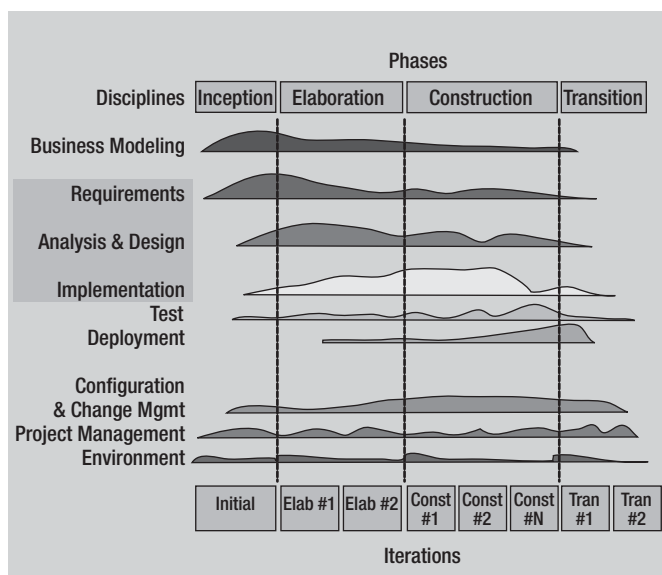
De Rational Unified Process (RUP) methodiek voor ontwikkelprojecten is gebaseerd op Object-Oriëntatie, Unified Modeling Language (UML) en Use Cases en daarom niet goed bruikbaar voor BI. Uit onderzoeken is gebleken dat Object-Oriëntatie en UML niet goed toepasbaar zijn op dit gebied. En een datawarehouse bijvoorbeeld is wel beschouwd meestal één enorme Use Case, waardoor geen enkele scheiding van functionaliteit of prioritering kan worden aangebracht.

Component voor component wordt toegewerkt naar de uiteindelijke rapportage of interface

Het gevolg van het ontbreken van een duidelijke methode is dat de risico's in BI-projecten niet worden onderkend en dat vervolgens een tijdsoverschrijding of budgetoverschrijding dreigt. Doordat de gebruikersfunctionaliteit aan het einde van het project wordt gerealiseerd is het risico van ontbrekende of verkeerd geïmplementeerde functionaliteit groot. Ook komt het geregeld voor dat op het laatste moment functionaliteit wordt geschrappt wegens tijdsdruk en dan met name in de laatst ontwikkelde gebruikerstoeegang, zoals de OLAP-kubussen of rapportages. De gebruikers van het BI-systeem zijn altijd de grootste verliezers: Zij worden niet alleen voor deze oplopende kosten en vooruitgeschoven oplevering gesteld, maar zij kunnen vervolgens ook minder waarde onttelen aan de uiteindelijke oplossing. Ook blijkt vaak dat er een ander datawarehouse of BI-oplossing is gerealiseerd dan dat de organisatie in gedachten had.

RUP for BI

Hoe nu te komen tot een volwaardige methodiek voor BI-projecten? In reguliere ontwikkeltrajecten heeft men de risico's beperkt en de voorspelbaarheid vergroot door het toepassen van methodes als DSDM en RUP. Op basis van deze ervaring is een BI-variant op RUP ontwikkeld, die niet meer afhankelijk is van Object-



Afbeelding 1: Overzicht van de noodzakelijke aanpassingen aan RUP om BI-projecten mogelijk te maken.

Oriëntatie. De aanpassingen in RUP die hiervoor nodig zijn, worden inzichtelijk gemaakt in afbeelding 1. Vanuit met name de disciplines Requirements, Analyse & Design en Implementation is de Object-Oriëntatie vervangen, zonder de onderliggende principes uit het oog te verliezen. Deze principes zorgen er namelijk voor dat het ontwikkelproces goed verloopt. De belangrijkste nieuwe mechanismen binnen RUP voor BI zijn:

- Splitsing van de functionaliteit in Data Flows;
- Het gebruik van BI Models voor prototyping;
- Verdeling van de componenten in Technical Flows.

Aan de RUP methodiek zijn natuurlijk ook specifieke BI-procesen, activiteiten en producten toegevoegd.

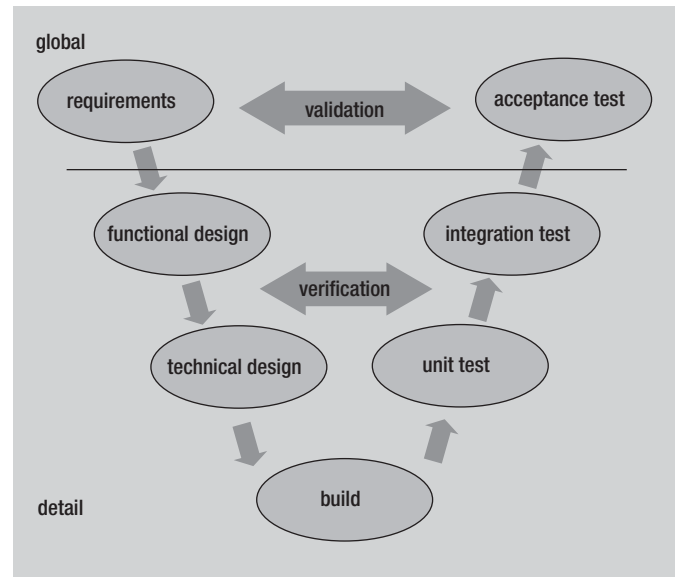
Splits de functionaliteit op

In een kenmerkend BI-project wordt aan het begin een compleet functioneel ontwerp opgesteld en vervolgens wordt dit volledig uitgevoerd. Hiermee wordt de waterval-methode gevolgd en is er geen scheiding van functionaliteit. Op deze manier ligt alles op het kritieke pad en blijkt pas dat bepaalde mechanismen niet werken als de betreffende fase in het project is bereikt. Bij een datawarehouse-project voor de financiële en personele administratie van een dienstverleningsbedrijf stuitte men twee weken voor de oplevering op een groot probleem. De ontwikkeling van het ETL-proces voor de moeilijkste dimensie 'Employee' was nog niet uitgevoerd en dit bleek technisch ook niet mogelijk te zijn. In het ontwerp was namelijk geen rekening gehouden met het feit dat uit de bron alleen dienstverbanden, in plaats van werknemers, werden aangeleverd. De aanpassing van het ontwerp zorgde in dat stadium voor een behoorlijke verschuiving van de planning. In de RUP-aanpassing zorgen Data Flows voor een opdeling van de functionaliteit, waardoor de requirements kunnen worden verdeeld. Vanuit de verdere uitwerking van de Data Flows kan een keuze worden gemaakt in de prioriteit. De moeilijkste functionaliteit of het nieuwe technische mechanisme dient nu als eerste uitgevoerd te worden en dus al in de uitwerkingsfase 'Elaboration Phase' te worden gerealiseerd. Zo kan in een vroeg stadium worden aangetoond dat hetgeen bedacht is, ook echt werkt.

Gebruik prototypes

Nadat de eerste Data Flows zijn uitgewerkt, moet gestart worden met het visueel maken van dat wat de gebruikers van de BI-oplossing kunnen verwachten. De gebruikers komen zo al aan het begin van het project te weten welke functionaliteit het systeem gaat bieden. Standaard wordt aan het begin van een project namelijk een set systeemspecificaties opgesteld en na de validatie zijn er vrijwel geen mogelijkheden meer om aanpassingen hierin aan te brengen. Om via Change Requests de functionaliteit te wijzigen is meestal een dure aangelegenheid.

Prototyping zorgt voor veel meer betrokkenheid van de gebruikers bij het project. Via deze prototypes of BI Models kan een directe interactie plaatsvinden en kunnen aanvullende requirements worden gevonden. Daarnaast kan de gebruiker toetsen of hetgeen gewenst is ook door het project opgeleverd wordt. Mogelijkheden



Afbeelding 2: Het V-model voor testen. De opdeling in Technical Flows maakt het mogelijk om BI-oplossingen op alle niveau's te testen.

voor deze interactie zijn niet alleen datamodellen, maar juist ook OLAP-kubussen en dynamische rapporten. Niet alleen wordt de drempel voor acceptatie lager, ook wordt zo de basis gelegd voor technische ontwikkeling.

Deel het systeem op

De meeste datawarehouses of BI-oplossingen worden ontwikkeld als één groot systeem. Component voor component wordt toegevoerd naar de uiteindelijke rapportage of interface. De test uitgevoerd door de ontwikkelaars bestaat dan ook uit het uitvoeren van een integratietest op het volledige systeem. Nadat de taak van de ontwikkelaars volledig is verricht, is er alleen nog een acceptatietest mogelijk die weer als een integratietest wordt uitgevoerd. Bij het vinden van een fout ligt vervolgens de hele test stil totdat de fout is opgelost. Het geheel van ontwikkeling en testen is zo een moeilijk te plannen proces met enorm veel afhankelijkheden en risico's. Bij een Europese bank is voor iedere week ontwikkelwerk ongeveer een maand testwerk nodig. Omdat iedere wijziging in het datawarehouse als één set software wordt opgeleverd, zijn meestal vijf of zes testcycli met productiedata noodzakelijk om acceptabele resultaten te verkrijgen.

De gebruikers van het BI-systeem zijn altijd de grootste verliezers

In de RUP voor BI-aanpak wordt het systeem technisch verdeeld in een aantal Technical Flows. Deze kunnen gerealiseerd worden zonder dat voorafgaande onderdelen volledig gereed zijn. De afhankelijkheden tussen deze Flows worden bewaakt in het inte-

gratieplan. Door deze technische scheiding kunnen onderdelen van het systeem niet alleen apart worden ontwikkeld, maar ook getest, zoals getoond wordt in afbeelding 2. In de aanpak worden dan ook activiteiten ontplooid om iedere Flow te voorzien van testcases. De afzonderlijke Technical Flows (op basis van technical design) kunnen worden getest door middel van een unit test, gevolgd door een integratietest van een groep afhankelijke Flows. De acceptatietest kan natuurlijk ook worden uitgevoerd op basis van iedere afzonderlijke Flow, waarna de opvolgende implementatie geleidelijk kan plaatsvinden.

Passende methodiek voor BI-ontwikkelingen

Datawarehouse-projecten zijn moeilijk en vaak worden planning en budget overschreden. Door het toepassen van een geschikte methodiek wordt het proces geborgd. Hierdoor wordt de betrokkenheid van de eindgebruikers gegarandeerd en wordt het resultaat voorspelbaar. Als eenmaal de moeilijkste obstakels zijn

overwonnen, kan parallel worden gewerkt aan de volledige ontwikkeling.

Bij de bouw van een datawarehouse of toevoeging van nieuwe functionaliteit aan een bestaand datawarehouse, doet men er verstandig aan eerst een keuze te maken voor een geschikte ontwikkelmethodiek. Men weet vanaf de start waaraan men begint en is niet nodig achteraf te concluderen dat het project aanmerkelijk langer heeft geduurd en duurder is geworden en dat niet de gewenste functionaliteit is verkregen.

De beschreven RUP-variant maakt deel uit van Capgemini's geoptimaliseerde software engineering methodiek voor Business Intelligence, het ADC BI (Accelerated Delivery Center for Business Intelligence).

Jan-Paul Fillié

Jan-Paul Fillié (janpaul.fillie@capgemini.com) is consultantop het gebied van Business Intelligence bij Capgemini.