

De centrale rol van data, ook bij de overheid

Thijs van der Feltz (5-12-2019)

Bij architectuur hoort informatie & kennis op de voorgrond te staan, waarbij de medewerkers, klanten en het management centraal staan.

Ik zie niet in dat de overheid een andersoortige architectuur zou behoeven dan het bedrijfsleven. Hooguit dat de overheid misschien wat meer transparantie zou moeten geven richting de burger/belastingbetaler.

Meer toegespitst op data(architectuur) zie ik al helemaal geen verschil tussen overheid en bedrijfsleven. Qua datagebruik doet de overheid in essentie geen andere dingen dan het bedrijfsleven.

Ook de overheid zou klantgericht moeten zijn, zoals voor het bedrijfsleven 'know your customer' geldt. De overheid heeft één voordeel: een uniform BSN als koppelmechanisme voor verschillende administraties.

Meer aandacht voor data

Er zou een overkoepelende data architectuur moeten zijn voor de overheid d.w.z. dat ieder type object (bijv. natuurlijk persoon, rechtspersoon, gebouw, weg, stoplicht, voertuig) slechts één uniforme identifier moeten hebben, plus een set nauwkeurig gedefinieerde basiseigenschappen. Dit zou in delen ontwikkeld kunnen worden, per zo genaamde 'subject area' met daarbij een plattegrond die alle subject area's met elkaar verbindt. Met zo'n plattegrond kunnen verschillende overheidssystemen autonoom ontwikkeld worden, en toch vrij eenvoudig worden geïntegreerd v.w.b. gegevensvastlegging en uitwisseling. Uiteraard is data governance nodig om vast te stellen wie de verantwoordelijke 'eigenaar' van elk gegeven is, en wie de 'gebruikers' van gegevens zijn. Dit vormt de basis voor een publish/subscribe model.

Het CBS is overigens een interessant voorbeeld van (1) een partij die groot belang heeft bij (op basis van uniforme identifiers) connected data en (2) commerciële doelen kan nastreven door het integreren, verrijken en doorverkopen van (overheids)data aan o.a. het bedrijfsleven. Het CBS zit dus op het raakvlak tussen overheid en bedrijfsleven. Ook de Kamers van Koophandel (database) is zo'n voorbeeld.

Datamanagement

Meer serieuze aandacht is noodzakelijk voor DATA in het algemeen, en met name metadata. Zonder metadata (waaronder datamodellen en *data lineage*) heeft data geen context en daardoor ook geen waarde of zelfs valse waarde omdat het onjuist geïnterpreteerd kan worden.

Data (architectuur) overstijgt projecten en applicaties en dient dus op een andere manier ontworpen en gemanaged te worden. De praktijk is echter het oplossen van de ergste problemen d.m.v. 'datakwaliteitsprojecten', hetgeen niet meer is dan een pleister op de wond. Dit soort acties ontstaan vaak te laat, pas als een datakwaliteitsprobleem geconstateerd (zichtbaar/herkenbaar) wordt. Veel datakwaliteitsproblemen zijn onzichtbaar en dat is eigenlijk nog gevaarlijker omdat men analyses doet op data met een vals gevoel van zekerheid.

De drie belangrijkste pijlers voor succesvol datamanagement zijn: Data Architecture (de blauwdruk), Data Governance (wie doet wat) en Metadata Management (zonder

metadata is data überhaupt niet te managen want "One cannot manage what one cannot define").

Voorkomen is beter dan genezen. Dat geldt in hoge mate voor datakwaliteitsproblematiek. Dit wordt vaak uitgedrukt in de vorm van de 1-10-100 regel: het kost €1 om een fout bij data entry te herstellen, €10 tijdens het proces en €100 achteraf (na analyses, rapportages en besluitvorming).

Veel mislukte of vertraagde projecten hebben hun oorsprong in slecht data ontwerp. Omdat men vaak niet verder kijkt dan de eerste toepassing van data (voor één specifiek doel) komt men er steeds achter dat data ook in andere functies/situaties gebruikt wordt, maar dan net iets anders. Hierdoor moet het ontwerp steeds bijgesteld worden en moeten reeds voltooide onderdelen met terugwerkende kracht aangepast worden. Of nog erger, men doet een soort 'quick fix' *workaround*, hetgeen natuurlijk *het* recept is voor toekomstig onderhoudsproblematiek. Door vooraf alle gegevens vanuit een breed organisatieperspectief te analyseren en te ontwerpen, kunnen dit soort problemen grotendeels voorkomen worden. Dit kost initieel tijd in een project en wordt daarom niet of minimalistisch opgenomen in de planning. Op termijn bespaart dit een hoop ellende, moeizame discussies en paniekvoetbal omdat de deadline herhaaldelijk overschreden wordt.

Databeheer

Organisaties zouden er goed aan doen om een aparte afdeling of functie op te zetten, uitsluitend voor beheer en governance van data als een op zich zelf staande 'corporate resource' (of 'asset', zoals andere financiële middelen). Een soort Data Management/Information Governance Office o.i.d. Zonder dit heeft men totaal geen overzicht van de beschikbare data(bronnen), hetgeen resulteert in meervoudige (inconsistente) kopieën van dezelfde gegevens, van onbekende kwaliteit. Ook het vinden van data wordt hierdoor zeer moeilijk, hetgeen leidt tot veel extra werk en redundante data. Onbeheerste redundantie is kostbaar vanwege duplicatie van werk, opslag en synchronisatieprocessen; en de uitdaging van consolidatie van (bijna) onverenigbare gegevens.

Tenslotte dient datakwaliteit de verantwoordelijkheid te zijn van de business en niet van IT. Gemakshalve wordt dit probleem in de praktijk te vaak doorgeschoven naar IT als een technisch dingetje, maar de realiteit is dat de business zowel eigenaar als gebruiker van alle data is, en daarbij ook alle lusten & lasten ervan draagt. IT is slechts faciliterend, niet veel meer dan een 'doorgeefluik' als het gaat om data. Opslag van data is het minst interessant; bij data/informatie gaat het vooral om semantiek, taal, context en communicatie; tussen business onderling en tussen business en de buitenwereld.

Projectsucces

Lessen uit het verleden hebben ons geleerd dat het succes van een project altijd begint met goede/heldere/concrete requirements. En daar gaat het al vaak mis. Een goed uitgewerkt en afgestemd datamodel biedt hoge precisie in expliciete(!) requirements, hetgeen van cruciaal belang is voor het leggen van een breed architectuurfundament. Helaas lijkt deze essentiële succesfactor een 'verloren vaardigheid' te zijn, die gemakshalve (of door onwetendheid) wordt overgeslagen. Er is een grote correlatie tussen projectsucces en de aanwezigheid van een robuust datamodel. Een datamodel is de belangrijkste component van de data-architectuur. Het is zeer tijdbestendig, veel meer dan processen en applicaties. Alleen daarom al is het een zinvolle investering, ook (keer op keer) voor transitie en migraties naar

nieuwe applicaties. Ook biedt een datamodel een gemeenschappelijk begrippenkader, hetgeen noodzakelijk is om de kloof tussen business en IT te overbruggen, om hierdoor de business (= ook de overheid) beter te kunnen bedienen (en minder geld te verspillen aan miscommunicatie).

Kleine teams

Ik ben wel een voorstander van relatief kleine teams, die op basis van een goede (data)architectuur goed op elkaar afgestemd zijn. De complexiteit van het managen van teams neemt exponentieel(?) toe met het aantal deelnemers vanwege de 'ruis' (overhead) en de communicatieproblemen. Ook is het meer motiverend voor mensen om in een klein(er) team aan iets concreets te werken. Men staat dan dicht bij het eindproduct en voelt zich daarbij meer betrokken en verantwoordelijk voor de kwaliteit ervan.

De kunst van kleine teams vormen heeft natuurlijk alles te maken met het goed opdelen van grote projecten/programma's. Dit vereist diepe architectuurkennis, om complexe systemen zinvol op te splitsen in "loosely coupled" componenten. Goede kennis van de onderliggende data-architectuur (waaronder ook de data lifecycle en -eigenaarschap) kan hierbij een groot hulpmiddel zijn tot het vormen van (tijdbestendige) logische functionele blokken (=deliverables) die in correcte chronologische volgorde ontwikkeld worden.

Zelf doen of uitbesteden

Probleem met uitbesteden is dat veel kennis verdwijnt als de mensen vertrekken. Documentatie is vaak een sluitpost waarop bezuinigd wordt, dus interne mensen met "legacy" kennis zijn noodzakelijk en goud waard. Helaas worden deze mensen vaak ondergewaardeerd en onderbetaald. Outsourcing is prima voor 'hapklare' stukken ontwikkelwerk met heldere specificaties (zowel 'functional' als 'non-functional' requirements), maar de opdrachtgever moet te allen tijde de regie behouden en voldoende expertise hebben om de resultaten op kwaliteit te beoordelen. Ik heb dit vaak mis zien gaan bij bijv. uitbesteding naar India. Die mensen staan onder tijdsdruk voor snelle resultaten, maar vaak gaat dit ten koste van elegantie en onderhoudbaarheid van de software (lees: spaghetti-code), hetgeen een goede garantie is voor vervolgoopdrachten voor zo'n bedrijf i.v.m. ingewikkeld onderhoud. Sleutelfiguren moet je dus zoveel mogelijk in huis houden, ook vanwege continuïteit en bewaking van lange termijn (architectuur) doelen.

Vasthouden talent

De overheid is doorgaans zuiniger met salarissen dan het bedrijfsleven, waardoor het te weinig toptalent aan kan trekken. Hierdoor is het noodzakelijk om dure externen in te huren, die hun kennis en ervaring na afloop weer met zich meenemen. Mooi voorbeeld van goedkoop=duurkoop. Dit is echter makkelijker gezegd dan gedaan, want het vinden en herkennen van talent is veel moeilijker dan het aftikken van pro forma eisen, zoals Togaf-certificatie. Het vereist diep inzicht in de aard van verschillende soorten architectuurfuncties om het nodige talent in een ander te herkennen. Of, zoals de Engelsen het bondig zeggen: 'It takes one to know one', een variant op 'je weet niet wat je niet weet'. Kortom, een soort kip/ei cirkel in talent scouting die doorbroken moet worden.

Een praktijkvoorbeeld: Vacaturebeschrijvingen voor bijvoorbeeld (data)architectuur functies zijn vaak niet meer dan een lange lijst van kreten, hetgeen aangeeft dat zo'n organisatie geen goed beeld heeft van wat het nodig heeft op dit gebied. Een

talentvolle architect herkent dit en zal zich terecht afvragen of er wel eer en resultaat te behalen is bij zo'n organisatie.

Tenslotte: zorg voor een goed multidisciplinair team van ervaren architecten met invloed om het IT beleid te formuleren, bij te stellen en te bewaken.

Hierbij wil ik het belang van data architectuur extra benadrukken omdat:

- met name data architectuur (ten opzichte van applicaties/technologie) de meest langlopende gevolgen heeft op IT, met derhalve een enorm ROI potentieel.
- deze discipline doorgaans de meest onderbelichte (en minst begrepen) vorm van architectuur is, en hard toe is aan een inhaalslag.