

De -eenvoudige- basis van IT-architectuur

Een vakantieplan

Stel, je wilt samen met je gezin of met vrienden op vakantie. Je weet wie er mee willen, je weet dat de vakantie ergens in juli zal plaatsvinden en je wilt samen een leuke tijd hebben. Verder is er nog niets concreets bekend. Bij de gezamenlijke bespreking vliegen de ideeën in het rond, van een cruise naar de poolcirkel tot een all-in resort in Ivoorkust en een trektocht langs het Pieterpad. Twee van de deelnemers bespreken ondertussen merken van slaapzakken en twee anderen wisselen tips uit over wagenziekte.

Je besluit om de aanpak te structureren en maakt het volgende stappenplan waarin je van grof naar detail de besluitvorming doorloopt:

- Wat willen we bereiken, oftewel, wanneer is de vakantie geslaagd?
- Welke mogelijkheden/middelen gaan we dan inzetten op weg naar dat doel?
- Hoe vullen we die mogelijkheden en middelen dan concreet in?

Je besluit om straks, als deze vragen van boven naar beneden beantwoord zijn, nog een controleslag in te bouwen, namelijk:

- Zijn we nog iets vergeten? Klopt het geheel wat we nu bedacht hebben?

Vervolgens loop je rustig de drie vragen langs die je hebt opgesteld en daar komt het volgende uit.

Wat willen we bereiken, wanneer is de vakantie geslaagd?	<ul style="list-style-type: none">• We willen er met zijn allen op uit• Er moet gelegenheid zijn voor ontspanning: luieren, zonnen• Er moet ook activiteit mogelijk zijn, sportieve uitdagingen, maar niet iedereen zal daaraan meedoen• Zwemgelegenheid is belangrijk voor iedereen• We willen iets nieuws ontdekken• Mooi weer is belangrijk• We hebben een bepaald budget daar blijven we binnen
Welke mogelijkheden en middelen gaan we inzetten?	<ul style="list-style-type: none">• Liever geen vliegtreis.• Reizen met de auto is oké, maar per dag niet langer dan zes uur• Kamperen is te inspannend, een hotel te onvrij, misschien is ergens een huis te huren?
Hoe vullen we de mogelijkheden en middelen concreet in?	<ul style="list-style-type: none">• We reizen met de auto in drie dagen naar Zuid-Frankrijk waar we een ruim chalet gehuurd hebben, niet ver van zee.• Onderweg logeren we in hotels• Vanuit het chalet kunnen liefhebbers verschillende bergwandelingen maken.

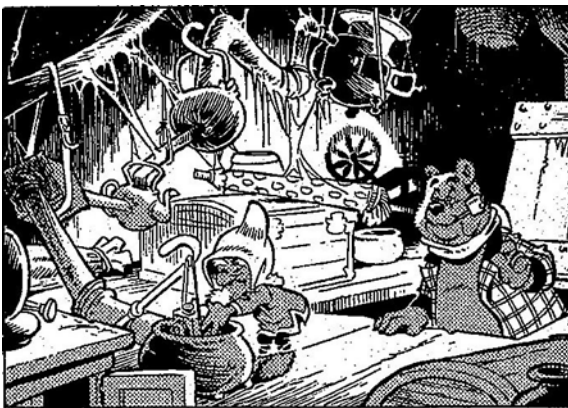
Deze uitkomst geeft invulling aan de gestelde doelen, en daarmee is het tijd voor de controleslag: zijn we nog iets vergeten? Bij het verder invullen van het plan blijkt dat er gekozen moet worden tussen ófwel de vakantie beginnen in de laatste week juni, ófwel een budgetoverschrijding accepteren van 500 euro. De laatste week juni zou betekenen dat twee mensen die eerste vakantieweek niet mee kunnen en dus wordt er besloten om de budgetoverschrijding te accepteren. Dan kan de heenreis gezamenlijk gemaakt worden.

IT-architectuur voor niet-IT-architecten

De aanpak in dit voorbeeld is precies dát wat IT-architectuur beoogt: op een gestructureerde manier vastleggen:

- Wat willen we bereiken, wat is het doel? (Het “waarom”/”waartoe” van de informatievoorziening)
- Wat gaan we dan gebruiken om dat doel te bereiken? (Het “wat”/”welke middelen” van de informatievoorziening)
- Hoe gaan we die middelen dan inzetten of tot stand brengen? (Het “hoe” van de informatievoorziening).

De basis van IT-architectuur is dus eigenlijk niets bijzonders. Die basis wijkt niet af van wat we normaal (met “boerenverstand”) altijd al doen. Het heeft immers geen zin om details van het “hoe” in te vullen (zoals wagenziektepillen) als het “waarmee” nog onduidelijk is (trektocht langs het Pieterpad).



Kwetal was juist bezig een eigenaardig doosje in een pot te hangen, toen heer Ollie boven kwam.

“Een kwibraan,” sprak het ventje trots. “Nu kan je hier wieling krijgen, zoveel als je wilt. Het was moeilijk om de kaggel daar te bevroezelen zonder een heffer.” “Ja, dat snap ik,” zei zijn gastheer zonder begrip. “Héél mooi.”

(Uit: De Kiekvogel, Marten Toonder, 1958)

Waarom IT-architectuur toch vaak ingewikkeld lijkt

Toch lijkt IT-architectuur vaak héél ingewikkeld. Er worden termen en begrippen gebruikt die alleen “ingewijden” lijken te beheersen. Zelfs tussen verschillende “ingewijden” ontstaan vaak misverstanden omdat de een met een bepaald begrip iets anders bedoelt dan de ander. En er komt vaak heel veel techniek om de hoek als er over de informatiearchitectuur gesproken wordt. Hoe kan dat als de basis zo eenvoudig is?

Gesprekken met-, maar ook tussen IT-architecten lijken vaak te verlopen als het gesprek van Ollie B. Bommel met Kwetal (zie inzet). En dat gesprek vond ruim plaats vóór het IT-tijdperk.

Dat “IT-architectuur” vaak heel ingewikkeld lijkt komt (onder andere) door:

- Het aanzien van “constructie” voor architectuur
- Het onvoldoende begrijpen van de eigenlijke vraag (het “waartoe”) door IT-architecten
- Het feit dat we nu eenmaal te maken hebben met veel IT-voorzieningen die niet onder een adequate architectuur tot stand zijn gekomen.

IT-architectuur voor niet-IT-architecten

Een korte toelichting voor we verder gaan met hoe het wél moet.

Het verschil tussen “constructie” en “architectuur”

“Constructie” gaat over “hoe maak ik het”. Architectuur gaat over “hoe hangt het samen”. Daartussen zit verschil.

Vanuit architectuur zeg je “ik heb een brug nodig over een rivier. Er moet een vaargeul vrij blijven van voldoende hoogte en het rijdek mag niet te hoog liggen want er is weinig ruimte voor de op- en afritten. Er moet een tweebaans autoweg over komen plus een vrijliggend fietspad en voetgangerspad. Een boogbrug zou mooi zijn.

Vanuit constructie zeg je dan “oké, dat kan dan een vrijdragende betonnen brug worden, dat is geen boogbrug maar het voordeel is dat er geen pijlers nodig zijn. Of een klassieke metalen boogbrug met twee betonnen pijlers waartussen maximaal 250 meter vrij blijft. Voor het metaal gebruiken we <en dan volgen de constructiedetails> de boog wordt <constructiedetails> en een betonnen brug zouden we als volgt construeren < constructiedetails>.

Een probleem in de IT-architectuur is dat er vaak heel gedetailleerd gesproken wordt over constructiedetails (“Java/C++, module XYZ, product Q, protocol zus-en-zo, we doen het met blockchain”) terwijl de architectuur zeer beperkt wordt ingevuld (hoe is de samenhang, welke afhankelijkheden zijn er, in welke omgeving moet het functioneren, op welke wijze draagt het bij aan het doel”).

De kans op missers is dan groot. (Er zijn tenten en wandelschoenen aangeschaft, maar we gaan op een cruise).

Onvoldoende begrip van de eigenlijke vraag

Om te begrijpen voor welk probleem we een IT-oplossing nodig hebben is er een daadwerkelijk *begrip van de vraag* nodig. Dat begrip is voor buitenstaanders lastig op te brengen. Een voorbeeld: eind vorige eeuw was een constatering dat de processen in Justitiële inrichtingen (“gevangenissen”) eigenlijk volstrekt vergelijkbaar waren met processen in het hotelwezen. Er is toen voor een aanzienlijk bedrag een Amerikaans systeem aangeschaft dat daar veel in hotels gebruikt werd. Met beperkte aanpassingen zou het geschikt gemaakt worden voor de Dienst Justitiële Inrichtingen in Nederland. Dat mislukte omdat er simpelweg toch véél te veel verschillen waren. Ook veel andere mislukkingen in de informatisering zijn toe te wijzen aan onvoldoende begrip van de behoefte, de vraag die gesteld wordt. Vaak in combinatie met een veel te optimistische kijk op aanpassingen in organisaties en werkprocessen “omdat het wegens de IT niet anders kan”.

Vooraf het helder maken van de vraag is dus het zwakke punt in IT-projecten, niet het bedenken en realiseren van de oplossing. Een goede architectuur-aanpak kan dat voorkomen, maar dan moet die architectuur wel vanuit “de business” gevoed worden. De mensen die het werk doen, die de organisatie kennen moeten de belangrijkste input geven, niet de “generieke IT-architect” of “adviseur” want die ziet vaak belangrijke verschillen in de manier van werken (en de omgeving) over het hoofd. Als er gebruik wordt gemaakt van klankbordgroepen moeten daar *inhoudelijk deskundigen* uit de betreffende organisatie in zitten en naar hen moet geluisterd worden.

Dit betekent ook dat het niet handig is om niet-IT-architecten tot in details te leren hoe je IT-architectuur bedrijft, het is veel belangrijker om wél-IT-architecten onder te dompelen in de organisatie waarvoor zij werken. IT-architecten moet helder en duidelijk kunnen communiceren in de taal van de (doel) organisatie. Kunnen zij dat niet dan liggen de misverstanden lagen dik op de loer. Voor niet-IT-architecten is het voldoende als zij goede vragen kunnen stellen en niet rusten voordat zij er naar tevredenheid antwoord op hebben gehad.

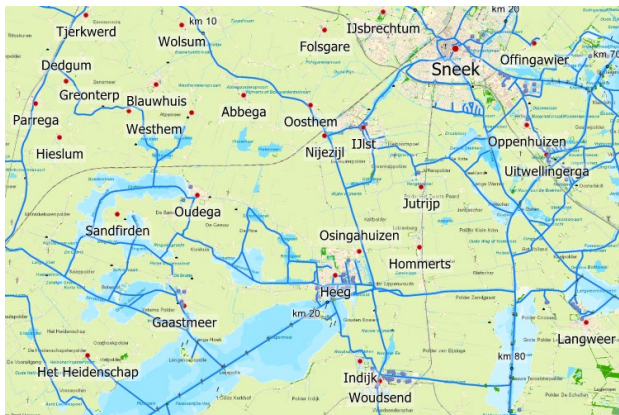
Veel ICT-landschappen zijn eerder “gegroeid” dan “ontworpen”

We komen uit een tijd waarin IT-architectuur sterk onderbelicht was, zich vooral richtte op de constructie (het “hoe”) en per “oplossing” werd ingericht. Een samenhangend, volledig en actueel overzicht over bijvoorbeeld het applicatielandschap, inclusief de manier waarop dit de verschillende werkprocessen en afdelingen in een organisatie ondersteunt is daarom vaak niet aanwezig. Toch willen we in zo’n “gegroeid” landschap nieuwe toepassingen ontwikkelen, technologie veranderen en aansluiten bij ontwikkelingen. Dat is een risico. Je kunt pas veilig iets veranderen als je begrijpt wát je verandert en hoe die veranderingen doorwerken in de omgeving.

Er zijn in deze gegroeide situaties veel valkuilen. Aan de ene kant wil je niet eerst álles tot in details in kaart brengen voordat je een verdere stap zet, dat zou namelijk betekenen dat je nooit verder komt. Aan de andere kant kun je ook niet op goed geluk een aanpassing doen en dan maar zien waar het schip strandt.

Met IT-architectuur kun je dan een middenweg kiezen waarbij je hele delen van de (onduidelijke) omgeving als “black box” beschouwt waarvan je alleen de output en input beschrijft. Op die manier kan IT-architectuur helpen om de risico’s te managen van de veranderingen.

“De” architectuur bestaat niet



Waterkaart en Wegenkaart naast elkaar (bron waterkaart: DKW Vaarkaart Nederland 2018, bron wegenkaart: www.viamichelin.nl)

Architectuur beschrijft samenhang en daarom bestaat “de” architectuur ook niet. Wat een bepaalde architectuur beschrijft hangt ervan af van de vraag wélke samenhang er inzichtelijk gemaakt moet worden. Als -bijvoorbeeld- een overheidsinstantie mensen door IT-modernisering aan de balie sneller wil helpen en daardoor per dag aan meer burgers haar diensten kan verlenen zal het aantal verkeersbewegingen naar die instantie toenemen: er komen meer burgers per uur aan en er rijden er meer weg. Hoort dan de capaciteitsaanpassing aan het parkeerterrein bij de (IT-)architectuur van die dienst? Of -andersom- het verminderen van de verkeerscapaciteit als belangrijke delen van de dienstverlening digitaal gaan worden uitgevoerd?

Invalshoeken (“viewpoints”)

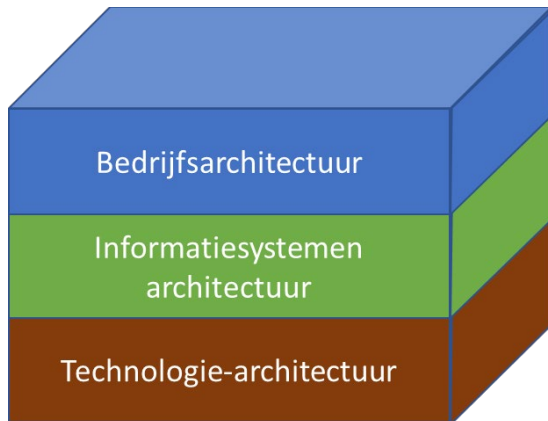
Bij IT-architectuur speelt, net als bijvoorbeeld in de cartografie, de belangrijke vraag “wát willen we inzichtelijk maken”. De onderstaande figuur toont naast elkaar een waterkaart (links) en een wegenkaart (rechts) voor hetzelfde gebied. De waterkaart toont de voor de scheepvaart belangrijke objecten, zoals bruggen, sluisen, waterdiepten en vaarroutes. Wegen, bebouwing en plaatsen zijn op de waterkaart “context” en worden weggelaten of minder nadrukkelijk weergegeven. De wegenkaart negeert

IT-architectuur voor niet-IT-architecten

waterdieptes, vaarroutes en de details betreffende bruggen en sluizen en toont juist wegen heel duidelijk, die worden zelfs qua breedte overdreven zodat ze er meer uitspringen.

Het zou onhandig zijn om één kaart te maken met en alle informatie voor de scheepvaart, en alle informatie voor het wegverkeer. Daarom wordt er gekozen voor een specifieke invalshoek: vanuit het gezichtspunt van een schipper is er andere informatie nodig dan vanuit het gezichtspunt van een automobilist.

Hetzelfde gebeurt bij (IT-)architectuur: er wordt een “viewpoint” gekozen, een invalshoek en vanuit die invalshoek wordt de architectuur beschreven. Er zijn drie invalshoeken die heel vaak gekozen worden en die samen een behoorlijk compleet beeld geven van het geheel en dat zijn¹:

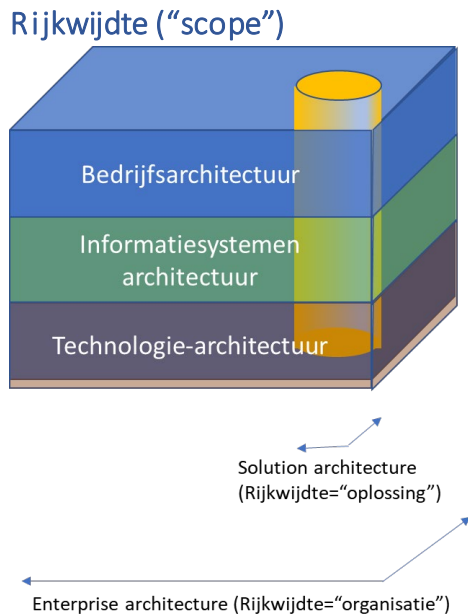


- De invalshoek vanuit de organisatie en de bedrijfsprocessen: de architectuur vanuit deze invalshoek noemen we de bedrijfsarchitectuur of in het Engels “business architecture”
- De invalshoek vanuit informatie en applicatiesystemen: de informatiesystemenarchitectuur of in het Engels “information systems architecture”. Let daarbij op het meervoud bij informatiesystemen/information systems!
- De invalshoek vanuit de techniek: de technologie-architectuur (technology architecture).

De volgende paragrafen gaan steeds op een van deze “architectuurlagen” in, maar eerst nog iets over de reikwijdte van architecturen.

¹ De termen zoals hier gebruikt komen overeen met de terminologie uit de ToGaftm architectuurmethode. Er zijn verschillende andere architectuurmethoden die enigszins andere termen gebruiken, maar de kern komt over het algemeen goed overeen.

IT-architectuur voor niet-IT-architecten



Als je met de boot of de auto een tocht door Friesland maakt is een waterkaart of wegenkaart voor de hele wereld alleen maar lastig. Hetzelfde geldt voor architectuur. Als er een project wordt gestart rond de vastgoedregistratie bij een gemeente dan is de architectuur van de bevolkingsadministratie niet relevant. Als de Inspectie voor de Gezondheidszorg een digitaliseringsproject start, dan is de IT-architectuur van de Rijksdienst voor het Wegverkeer minder interessant. Kortom: hoewel uiteindelijk alles met alles samenhangt moeten we architecturen beperken tot datgene wat voor ons doel van belang is.

Vanuit deze praktische overweging maken we onderscheid tussen:

Enterprise architectuur. Dit is architectuur die de gehele "enterprise" omvat, waarbij die "enterprise" een overheidsdienst, bedrijf of zelfs een stel samenwerkende organisaties kan zijn. Een enterprise architectuur omvat meestal alle drie genoemde invalshoeken, dus bedrijfsarchitectuur, informatiesystemenarchitectuur en technologie-architectuur.

Solution architectuur. Deze architectuur gaat over een specifieke (deel-) oplossing voor een bedrijfsprobleem. Bijvoorbeeld het realiseren van een nieuwe manier van werken in de meldkamer van de politie, samen met de introductie van een nieuw meldkamersysteem of het vervangen van de traditionele telefonische intake bij een GGZ-instelling door digitale aanmelding. Ook een solution architectuur omvat de drie invalshoeken bedrijfsarchitectuur, informatiesystemen-architectuur en technologie-architectuur. Helaas vindt er nogal vaak verwarring plaats tussen de (technische) ontwerpen van applicaties en solution architecturen. Ten eerste zijn ontwerpen meestal gericht op constructie en niet op samenhang (zie de paragraaf "Het verschil tussen constructie en architectuur") maar erger is dat deze ontwerpen de bedrijfsarchitectuur niet beschrijven. De samenhang tussen organisatie/bedrijfsprocessen en de te realiseren "oplossing" blijft dan onduidelijk en daarmee de kans op mislukking: de applicatie werkt, maar is niet (goed) bruikbaar.

Deel-architecturen. Hiermee wordt een architectuur bedoeld die noch de gehele enterprise, noch de gehele solution beschrijft. Een deel-architectuur kan ook beperkt zijn tot één of twee invalshoeken.

Referentie-architectuur

IT-architectuur voor niet-IT-architecten

Een bijzondere eend in de bijt is de referentie-architectuur. Terwijl IT-architecturen vrijwel altijd gaan over concrete organisaties, concrete problemen en concrete IT-voorzieningen of -projecten is een referentie-architectuur een model, een voorbeeld.

In het Franse staatsarchief wordt nog steeds een platina staaf bewaard van exact 1 meter lengte. Dit was jarenlang de referentie waarvan andere maat-staven konden worden afgeleid. Een referentie-architectuur heeft een vergelijkbaar doel: door de eigen architectuur (enterprise-, solution-, of dealarchitectuur) af te leiden van een referentiearchitectuur weet je dat je past binnen een gesteld kader.

Een hele belangrijke referentie-architectuur voor de Nederlandse overheid is de NORA (Nederlandse Overheid Referentie Architectuur). Dit initiatief dat al rond 2000 startte heeft geleid tot een groot aantal NORA “dochters” die, specifiek op een bepaalde sector gericht, de referentiearchitectuur-rol vervullen.

Een belangrijk voordeel van het volgen van een referentiearchitectuur is dat de aansluiting bij ketenpartners en collega-organisaties in dezelfde sector vaak goed is geregeld: je spreekt elkaars taal. Een ander belangrijk voordeel is dat er een basis is waarvan uit begonnen kan worden: verschillen tussen de eigen architectuur en de referentie-architectuur zijn in ieder geval reden tot nadenken. Een mogelijk nadeel ontstaat als “invullen van de referentiearchitectuur” belangrijker wordt dan “kijken hoe het bij ons echt zit”. Doel van architectuur is het inzichtelijk maken van samenhangen, niet het inkleuren van het plaatje van anderen!

De nu volgende hoofdstukken gaan specifiek in op de bedrijfsarchitectuur, informatiesystemenarchitectuur en technologie-architectuur. Aangegeven wordt welke input er gebruikt wordt, wat de inhoud is en welke output er geleverd wordt.

De beschrijvingen gaan uit van projecten waarin sprake is van een “bestaande situatie” en een “toekomstige” (beoogde) situatie. Dit is niet altijd het geval. Als een organisatie besluit om de huidige situatie in kaart te brengen om een begin te maken met “enterprise architectuur” is er geen verschil tussen de bestaande en toekomstige situatie. De toekomstige situatie zal in dat geval niet in de architectuur voorkomen.

De bedrijfsarchitectuur – “waarom en waartoe”

Bronmateriaal

De bedrijfsarchitectuur geeft aan waarom we iets doen en wat ermee bereikt moet worden. Belangrijke input komt uit:

- Het missie/visie statement van de organisatie
- Het bedrijfsplan/ondernemingsplan
- Beschrijvingen van noodzakelijke veranderingen in de organisatiestructuur of bedrijfsprocessen
- Eventuele probleemrapportages, knelpuntenanalyses, verbeterplannen

Inhoud

- Beschrijving van de bestaande organisatiestructuur, bedrijfsprocessen, bedrijfsobjecten², informatieobjecten, kengetallen, KPI's. Dit is de “bestaande bedrijfsarchitectuur” (in het Engels: “baseline business architecture”) Als informatie over *de bestaande situatie* niet beschikbaar

² Onder “bedrijfsobject” verstaan we alle relevante entiteiten bij de uitvoering van het bedrijfsproces. Zoals burger/klant/patiënt, vastgoed (Kadaster), medicatie (Zorgsector), pakket (Logistieke dienstverlening).

IT-architectuur voor niet-IT-architecten

is zal een inventarisatie nodig zijn maar die heeft geen volledigheid tot doel, alleen volledigheid voor zover relevant voor het te ondernemen plan.

- Beschrijving van de beoogde organisatiestructuur, bedrijfsobjecten, informatieobjecten, kengetallen en KPI's. Dit is de "doel-bedrijfsarchitectuur" (in het Engels: "target business architecture"). Deze architectuurbeschrijving is heel belangrijk omdat erin wordt aangegeven hoe de organisatie met de nieuwe/gewijzigde informatievoorziening als geheel zal functioneren.
- Beschrijving van het verschil tussen de bestaande en de doel bedrijfsarchitectuur. Deze "delta" zal in het project (of de projecten) ingevuld moeten worden.

Output

De Bedrijfsarchitectuur en een lijst van openstaande punten/risico's en aandachtspunten voor het vervolg.

De informatiesystemenarchitectuur – "waarmee"

Input

- De Bedrijfsarchitectuur. Deze geeft namelijk de scope aan waarbinnen de informatiesystemenarchitectuur wordt uitgewerkt.
- Eerdere architectuurdocumenten, beschrijvingen van applicatiegebruik, specificaties van koppelingen/informatieuitwisselingen enz.
- Probleem-meldingen en -analyses uit servicedesk-rapportages, op applicaties gerichte verbeterplannen, release planningen.

Inhoud

- Beschrijving van het bestaande applicatielandschap en de uitwisselingen tussen die applicaties. Niet beperkt tot alleen de digitale uitwisseling ("netwerk-koppelingen") maar ook beschrijvend welke informatie handmatig (door mensen) wordt overgetypt/via CD's wordt uitgewisseld of op andere wijze wordt overgenomen. De gebruikersinterfaces (GUI's) van applicaties zijn dus ook van belang. Dit is de "bestaande informatiesystemenarchitectuur" ("baseline information systems architecture"). Ook hier geldt dat als informatie over de bestaande situatie niet beschikbaar is een inventarisatie nodig is die alleen volledig moet zijn voor zover relevant voor het te ondernemen plan.
- Beschrijving van de te realiseren applicatiearchitectuur. Beschrijving van de in-scope informatiesystemen, de door hen ondersteunde processen/verwerkingen en de door hen verwerkte informatie. Dit is de doel-informatiesystemen-architectuur ("target information systems architecture").
- Beschrijving van het verschil tussen de bestaande en de doel informatiesystemenarchitectuur. Deze "delta" zal in het project (of de projecten) ingevuld moeten worden.

Output

De Informatiesystemenarchitectuur en een lijst van openstaande punten/risico's en aandachtspunten voor het vervolg.

De technologie-architectuur – "hoe"

Input

- De Bedrijfsarchitectuur (als context) en de Informatiesystemen architectuur (als scope)

IT-architectuur voor niet-IT-architecten

- Beschikbare netwerkconfiguraties, server-configuraties, configuraties van virtuele servers en/of containers, software licenties, gebruikscijfers, contracten met Cloud leveranciers enz.

Inhoud

- Beschrijving van de bestaande toegepaste technologie (servers, netwerken, Clouddiensten enz) ter ondersteuning van het in de Informatiesystemenarchitectuur beschreven baseline-applicatielandschap. Dit is de bestaande technologie-architectuur (“baseline technology architecture”). Ook hier geldt weer dat als informatie over de bestaande situatie niet beschikbaar is een inventarisatie nodig is voor zover relevant voor het te ondernemen plan.
- Beschrijving van de toekomstig toe te passen technologie (servers, netwerken, Clouddiensten enz) ter ondersteuning van het in de Informatiesystemenarchitectuur beschreven target-applicatielandschap. Dit is de doel-technologie-architectuur (“target technology architecture”).
- Beschrijving van het verschil tussen de bestaande en de doel technologie-architectuur. Deze “delta” zal in het project (of de projecten) ingevuld moeten worden.

Output

- De Technologie-architectuur en een lijst van openstaande punten/risico’s en aandachtspunten voor het vervolg.

Samenvatting

IT-architectuur komt in de kern neer op het gestructureerd oplossen van een behoefte. Die behoefte komt altijd voort uit een probleem of verandering in de werkelijke wereld. IT-architectuur *lijkt* soms ingewikkeld omdat technische details van de IT-oplossing worden verward met het helder maken van de vraagstelling en het goed definiëren van de randvoorwaarden waaraan de oplossing moet voldoen.

Het toepassen van architectuur is dus eigenlijk gewoon toepassen van gezond verstand volgens een bepaald recept.

Architectuur gaat over het inzichtelijk maken van samenhang (en daarmee ook het overzicht krijgen over consequenties). Omdat inzicht en overzicht tegelijkertijd beperking vereisen in diepgang, volledigheid of reikwijdte maakt de IT-architectuur gebruik van invalshoeken. De drie, steeds weer toegepaste, invalshoeken zijn: bedrijfsarchitectuur, informatiesystemenarchitectuur en technologie-architectuur. Deze drie architecturen zijn algemeen bekend en hebben een inhoud die tussen de verschillende architectuurmethoden grotendeels overeenkomt.

Qua reikwijdte onderscheiden we:

- de enterprise architectuur, die de gehele organisatie omvat en alle drie de invalshoeken
- de solution architectuur, die qua scope overeenkomt met de oplossing van één bedrijfsprobleem en die alle drie de invalshoeken omvat
- deelarchitecturen die willekeurig gekozen kunnen worden, zoals bij de vervanging van een technische component of een nieuwe versie van een al in gebruik zijnde toepassing.

Referentie-architecturen, zoals de NORA zijn (beproefde) voorbeelden. Het helpt om bij een referentiearchitectuur aan te sluiten omdat daarmee uitwisseling (ook ervarings-uitwisseling) met collega-organisaties en ketenpartners makkelijker wordt en omdat de eigen architectuur daarmee al een begin heeft. Het volgen van een referentiearchitectuur mag echter niet beperkt zijn tot “lippendienst”, dat leidt alleen maar tot extra verwarring.