

IT-architecturen hebben een slecht imago en IT-architecten worden gewantrouwd als het gaat om kostenbesparing. Dat slechte imago is verklaarbaar, zeggen **Joost Bakker, Frank Hendriksen en Matthijs Maat**. Architecten en opdrachtgevers zijn niet goed in staat te bepalen wat ‘genoeg architectuur’ is. Een architectuur is geen doel op zich. Architecten moeten beter worden aangestuurd.

In veel organisaties is IT-architectuur synoniem voor verspilling. Dat is geen handig imago in tijden waarin kostenreductie het belangrijkste – en vaak enige – agendapunt is. Resultaat is dat architecten niet bij kostenbesparings-initiatieven worden betrokken. Dat is jammer, want een goede architect legt de relatie tussen organisatiedoelstellingen en middelen zonder zich te beperken tot belangen van afdelingen of groepen medewerkers. Een bruikbare architectuurbijdrage ontstaat echter niet vanzelf. Sturing is noodzakelijk, door architect én opdrachtgever.

Het slechte imago van een IT-architectuur is verklaarbaar. Architectuurinspanningen hebben de neiging zich in zichzelf te verliezen, waarbij architecten en opdrachtgevers bepaald niet geremd worden door veelgebruikte architectuurmethodes, architectuurraamwerken en beschikbare referentiearchitecturen. Architectuurtrajecten lijden onder syndromen die vaak ook nog in combinatie voorkomen. De volgende zijn te onderscheiden:

- alles-moet-beter-syndroom;
- het-moet-simpel-syndroom;
- het-moet-mooi-syndroom;
- ik-moet-het-precies-weten-syndroom;

Architectuur blijft ondergeschikt aan bedrijfsdoelen

■ **het-moet-flexibel-syndroom.** Het meest zichtbare effect van deze syndromen is dat het architectuurwerk uit de hand loopt. Er wordt lang gewerkt aan ‘het opstellen van de architectuur’ zonder dat er (tussen)resultaten zijn waar iets mee kan worden gedaan. Het tweede effect is dat er veranderingen worden ingezet die te ambitieus zijn. Vaak is er sprake van diepte-investeringen in de beginfase, die pas rendement opleveren als het mooie vergezicht realiteit is geworden. Dat vergezicht wordt echter zelden bereikt. In organisaties die dit een paar keer hebben ervaren krijgt architectuur al snel een negatief imago.

Besparingen

Het gevolg is dat kostenbesparingen niet onder, maar zonder architectuur worden ingezet. Veel besparingsstrategieën zijn gericht op het verlagen van de IT-kosten. Daarmee worden kansen gemist, want zelden denkt men in besparingen die bereikt kunnen worden met méér IT. Bovendien zijn besparingen vaak suboptimaal. Aangezien ieder bedrijfsonderdeel een bijdrage moet leveren aan het kaasschaven, bespaart iedereen binnen de lokale invloedssfeer. En dit

herhaalt zich vervolgens weer binnen alle deelafdelingen. Op deze manier kunnen waardevolle initiatieven – bedrijfsbreed gezien – geremd of zelfs gestopt worden, terwijl inefficiënties juist blijven bestaan (zie Automatisering Gids 15 mei 2009: ‘Slim snijden en slim investeren’). Want wie heeft nog het overzicht over de gevolgen voor het geheel? Wie brengt de structurele keuzes in kaart? Een goede architect kan de relatie leggen tussen doelstellingen en middelen zonder zich daarbij te beperken tot belangen van afdelingen of groepen medewerkers. Het ligt dus voor de hand om architecten te betrekken in die besparingsvraagstukken. Waar is stroomlij-

ning van het IT-landschap mogelijk om zo onderhoudsinspanningen te beperken? Welke ontwikkel- en productieplatformen zijn geschikt om de benodigde applicaties op te realiseren? Welke bedrijfsactiviteiten kunnen gebruik maken van één gezamenlijke IT-dienst, zonder in de knel te komen? Welke alternatieve IT-architecturen kunnen afhankelijkheid van dure licenties voorkomen? Wat is een goedkoop, maakbaar en werkbaar alternatief om die noodzakelijke functionaliteit te realiseren? Geen vragen waar een goede architect van in de stress schiet. Die gaat aan de slag, scherpert de doelstelling/vraag aan,

werkt scenario's uit, schetst de consequenties per belanghebbende en geeft zijn visie op het 'beste' alternatief.

Spelregels

Maar zelfs met een goede architect gaat dit niet altijd vanzelf goed – de eerder genoemde syndromen liggen op de loer. Om een architectuurinspanning bruikbare resultaten te laten opleveren, is sturing nodig. En dus óók het actief inrichten, aansturen en volgen van architectuurprocessen. Opdrachtgevers en architecten zijn immers net mensen: werken aan iets nieuws, groots en meeslepends is nu eenmaal leuker dan werken aan iets bestaands en kleins. Het

gevolg is dat architecten noch opdrachtgevers zich snel de vraag stellen wat ‘genoeg’ architectuur is. Voor de sturing van architectuur geldt een aantal belangrijke spelregels, waarbij een rol is weggelegd voor zowel opdrachtgevers als architecten:

■ Iedere architectuuractiviteit heeft een opdrachtgever

Sturing vergt een duidelijk doel. Architectuur is geen doel op zich en dus is bij sturing van architectuurinspanningen een opdrachtgever nodig. De opdrachtgever formuleert het doel en het type beslissing dat genomen moet worden op basis van de resultaten. De

architect formuleert de architectuuropdracht zo precies mogelijk en vraagt de opdrachtgever deze af te tekenen. De opdrachtgever kan – als daar aanleiding toe is – gerust de opdracht bijstellen. Het ‘maken van een architectuur’ is géén goede opdracht.

■ De architect adviseert, de opdrachtgever beslist

De architect levert architectuurscenario's waaruit gekozen kan worden. Zicht op motivatie voor – en consequenties van – de verschillende keuzemogelijkheden is nodig om inzicht te krijgen in de opties en om een weloverwogen besluit te nemen. De opdrachtgever kiest

Veel projectleiders zien op tegen discussies met architecten

vervolgens zelf, want voor architecten heeft een elegante en mooie oplossing soms meer aantrekkingskracht dan het bereiken van het hogere project- of organisatiedoel. Bovendien hebben ook architecten eigenbelangen en daar zit ‘zichzelf misbaar maken’ niet bij.

— Tips voor opdrachtgever —

Eis duidelijkheid van de architect

Het aansturen van architecten is lastig. Veel opdrachtgevers en projectleiders zien op tegen discussies met architecten vanwege hun inhoudelijke kennis. Dat is onnodig. Ook testers weten op hun vakgebied veel meer dan degene die ze aanstuurt, maar die komen echt niet weg met de vrijheden die sommige architecten zich permitteren. Tips voor de omgang met architecten zijn:

■ Plan het hoogst noodzakelijke architectenwerk

In moderne ontwikkelmethoden wordt niets gedaan dat niet direct bijdraagt aan het eindresultaat. Waarom zou architectuur een uitzondering zijn? Bepaal wat anderen nodig hebben van architecten, laat architecten de daarvoor benodigde activiteiten zo concreet mogelijk beschrijven, plan die activiteiten in korte iteraties en bepaal de voortgang. Dat geldt binnen ontwikkelprojecten: uiteraard is het niet verboden om voordat de eerste regel wordt ingetikt je een beeld te vormen van de grote lijnen van een systeem, maar werkzaamheden als ‘werken aan de architectuur’ zijn niet te plannen noch op voortgang te meten. Dat geldt echter zeker ook voor architectuurinspanningen die projecten overstijgen.

■ Eis dat architectuurscenario's realistisch blijven

Eis dat de scenario's uitgewerkt worden in termen die betrekking hebben op de huidige werkelijkheid. Wat verandert er voor wie? Wanneer? Waar in de tijd worden kosten gemaakt of besparingen bereikt? Prachtige vergezichten zijn aanlokkelijk, maar blijven in de praktijk vaak onbereikbaar. Architecturen die geen aandacht schenken aan de bestaande situatie en aan realistische stappen om die om te vormen naar de nieuwe situatie, zijn in de praktijk vaak waardeloos.

■ Accepteer geen onbegrijpelijke methoden en technieken

Architecten gebruiken soms speciale tools en technieken om complexe vraagstukken aan te kunnen. Deze leiden soms tot voor leken onbegrijpelijke beschrijvingen. Accepteer dit niet en eis dat de architect dit omzet in een voor betrokkenen begrijpelijke vorm. Accepteer geen abstracte concepten anders dan als toelichting op een scenario.

■ Stel de ‘what if’-vragen

Verschillende positieve en negatieve krachten maken projecten groter dan strikt noodzakelijk: enthousiasme, aanzien en ambitie, economisch belang, polderoplossingen. Architecten gaan daar vaak in mee en beschrijven de ideale eindsituatie. Maar wat als het tegengit? Als tijd en geld eerder op zijn dan dat die ideale situatie is bereikt? Is er dan toch iets bereikt dat waarde toevoegt of kosten bespaart? Eis dat architecten aantonen dat er afzonderlijke tussenstappen zijn die waarde toevoegen.

— Tips voor architecten —

Zelfsturing onontbeerlijk

Opdrachtgevers zijn vaak zeer terughoudend of niet in staat om de inhoud in te duiken en willen niet lastig gevallen worden met genuanceerde verhalen of complexe keuzes. Dat zadelt architecten op met een lastige klus: hoge verwachtingen, maar weinig hulp. Enige vorm van zelfsturing is onontbeerlijk. Tips daarbij zijn:

■ Betrek de opdrachtgever bij onzekerheid

Ga de inherente onzekerheid in architectuurtrajecten niet te lijf met papier en whiteboards alleen. Dat leidt onherroepelijk tot schijnzekerheid, zonder terugkoppeling uit de praktijk. Maak onzekerheden expliciet door het formuleren van aannames en zoek met de opdrachtgever naar mogelijkheden om die aannames snel te valideren. Een opdrachtgever mag duidelijkheid verlangen, maar geen helderziendheid.

■ Betrek de belangrijke belanghebbenden

Wie is de belanghebbende bij welke scenario's? Bespreek scenario's met de belanghebbenden zodat de consequenties herkend worden. Bespreek deze consequenties met de opdrachtgever en probeer de opdrachtgever de belanghebbenden vervolgens in de besluitvorming te laten betrekken, er is altijd iemand die pijn voelt.

■ Gebruik architectuurraamwerken selectief

Veel architectuurraamwerken beschouwen systemen vanuit verschillende invalshoeken. Dat is een krachtig middel, maar leidt vaak tot overcompleteheid. Gebruik de mogelijke invalshoeken zeer selectief en vul ze slechts in voor zover dat nodig is voor anderen om verder te kunnen. Is dat complete bedrijfs- of procesontwerp echt nodig?

■ Houd het tastbaar

Abstracties maken problemen behapbaar, maar de werkelijkheid soms ook onzichtbaar. Denk niet alleen in ‘logische informatiesystemen’ of ‘generieke functies’, maar probeer snel en regelmatig belangrijke elementen uit een architectuur concreet te maken en bijvoorbeeld te bewijzen met daadwerkelijke veranderingen of – binnen ontwikkelprojecten – met werkende code. Werk op alle abstractieniveaus tegelijk en niet stapsgewijs van een hoog naar een laag abstractieniveau.

■ Duidelijke kaderstelling voor architectuur

De opdrachtgever legt helder vast binnen welke termijn, financiële kaders en andere randvoorwaarden scenario's uitvoerbaar moeten zijn en zoekt bevestiging dat dit ook klopt. Bij omvangrijke vraagstukken stelt de opdrachtgever met de architect een multidisciplinair team samen dat de scenario's uitwerkt en waarin ieder vanuit zijn eigen vakgebied de realiseerbaarheid bewaakt. Architecten en opdrachtgevers moeten uitkijken architectuur zelf als kaderstellend te beschouwen – architectuur is en blijft te allen tijde een middel en daarmee

ondergeschikt aan bedrijfsdoelen. Strakke sturing op architectuur is nog geen gemeengoed. Het zal niet alleen wennen zijn voor opdrachtgevers, maar ook voor architecten. Willen ze daar niet aan? Stuur ze dan naar huis, dat levert in ieder geval een kostenbesparing op.

Joost Bakker, Frank Hendriksen en Matthijs Maat (matthijs.maas@mxl.nl) zijn werkzaam bij M&I/Partners, adviseurs voor management en informatie in Amersfoort (www.mxi.nl).

➤ Voor reacties en nieuwe bijdragen van deskundigen: Henk Ester (h.ester@sdu.nl, (070) 378 03 97).