

Het begrip "architectuur" heeft de afgelopen jaren veel aandacht gekregen in de ICT-wereld. En terecht. Nieuwe technologische ontwikkelingen en daarmee samenhangende veranderingen in de bedrijfsprocessen en de concurrentiepositie vragen om een lange-termijn visie. Voor concrete ontwikkelingen "onder architectuur" is visie alleen onvoldoende. We hebben ook architectuurproducten nodig die een concreet beeld geven van de ideale situatie en het eerst haalbare plateau.

achtergrond

Productgerichte architectuur

Op het niveau van de architectuurvisie is al veel gepubliceerd. Over architectuurproducten niet. In dit artikel willen we ingaan op de vraag welke architectuurproducten op welke architectuurniveaus nodig zijn als aanvulling op de "standaard" ontwerpproducten. We richten ons hierbij vooral op enterprise-architecturen.

ONGEDEFINIËRD BEGRIJF Wie de afgelopen jaren een seminar over architectuur bezocht heeft, zal zich verbaasd hebben over de verschillende en soms onverenigbare invalshoeken die de diverse sprekers kozen. In geschreven vorm zijn de verschillen niet minder. De ene auteur noemt alles architectuur wat structurend werkt in de applicaties, het ontwikkelproces of de ICT-organisatie, terwijl een ander juist tegen structureren en voor "beleven" is. Een derde heeft het alleen over een gelaagde architectuur voor applicaties en een vierde praat alleen over hardware en netwerken. Dit alles maakt het niet eenvoudiger voor mensen die concreet aan de slag willen in hun (klant)organisatie. Daar bovenop komt nog dat veel architectuurboeken en cursussen voor driekwart over analyse- en ontwerpmethoden gaan. Hierdoor wordt het wel erg moeilijk om de meerwaarde van architectuur ten opzichte van "gewone" ontwerpmethoden te doorgronden. Toch is die meerwaarde er wel degelijk.

De afgelopen decennia zijn er diverse analyse- en ontwerpmethoden ontwikkeld, zoals "Gestructureerde analyse en ontwerp" van onder meer Yourdon, "Information Engineering" van Martin en "Objectoriëntatie" van o.a. Booch, Jacobson en Rumbaugh. Deze methoden worden vooral toegepast voor het helder krijgen en implementeren van de func-

tionele eisen. Dit zijn eisen die gericht zijn op de directe toepasbaarheid, zoals "er moet een verzekeringsaanvraag geregistreerd en geaccepteerd kunnen worden".

De huidige architectuurgolf richt zich echter veel meer op belangrijke niet-functionele eisen die de business van nu stelt aan de organisatie en haar informatiesystemen. Voorbeelden zijn "snel nieuwe producten kunnen introduceren" en "eenvoudig aansluiten op nieuwe communicatiekanalen". De "gewone" analyse- en ontwerpmethoden bieden onvoldoende mogelijkheden om aan dit soort eisen tegemoet te komen. Aanvullend hierop zijn nieuwe technieken en vaardigheden nodig om tot een goede architectuur te komen. Want een goede architectuur bestaat niet uit een vaag verhaal, waarin veel goede wil getoond wordt, maar uit een set duidelijke en goed samenhangende producten, die richting geven aan ontwerpers.

ARCHITECTUURPRODUCTEN EN ARCHITECTUURNIVEAUS Binnen de ICT-wereld kan architectuur op verschillen niveaus worden bedreven. Per architectuurniveau stel je producten op die een duidelijk beeld geven van de doelen die je wilt bereiken en hoe die gerealiseerd moeten worden. De architectuurproducten van een business-architect zijn logischerwijs anders dan die van een software-architect. Toch is er ook overeenstemming te zien in de typen producten per niveau. En dat moet ook wel, want als het goed is, vormen de producten op een hoger architectuurniveau telkens het uitgangspunt voor de activiteiten en producten van de architect op het daaronder liggende niveau. Om alle producten in hun samenhang te kunnen plaatsen, beschrijven wij eerst de verschillende architectuurniveaus en daarna de typen architectuurproducten.

Tenslotte lichten we de producten per architectuurniveau toe.

ARCHITECTUURNIVEAUS Veel onnodige verwarring over het begrip architectuur is ontstaan doordat de spreker of auteur niet duidelijk maakt op welk niveau hij architectuur bedrijft. Want een principe, aanpak of

De huidige architectuurgolf richt zich op niet-functionele eisen die de business stelt aan organisatie en informatiesystemen

techniek op het ene architectuurniveau hoeft niet ook op andere architectuurniveaus te werken. Een indeling in architectuurniveaus en aspecten (een architectuurraamwerk, zie ook [IEEE] en [Zachman]) helpt bij het onderkennen van de architectuurgebieden die voor een organisatie belangrijk zijn. Ieder architectuurgebied heeft zijn eigen, specifieke problemen, principes, aanpak en producten.

De drie duidelijk van elkaar verschillende architectuurniveaus die we in dit artikel willen onderscheiden zijn business-architectuur, informatiesysteem-architectuur en software-architectuur.

1. BUSINESS-ARCHITECTUUR De business-architectuur moet een beeld geven van de eisen die de business op korte en lange termijn stelt aan de organisatie en de informatievoorziening. En van de keuzes die zijn gemaakt met betrekking tot de organisatie-inrichting en de informatievoorziening om aan deze eisen te kunnen voldoen. Centraal staat hier de visie van het management op de (door de ICT) veranderende markt en op de noodzakelijke veranderingen in de bedrijfsvoering. De business-architect begeeft zich op het grensvlak tussen organisatie-advies en ICT. Hij moet kennis hebben van beide gebieden, en de consequenties van wijzigingen in het ene gebied kunnen vertalen naar het andere.

2. INFORMATIESYSTEEM-ARCHITECTUUR De informatiesysteem-architectuur moet een beeld geven van de eisen die aan het totale informatiesysteem worden gesteld. De informatiesysteem-architect ontwerpt een indeling in deelsystemen zodat aan deze eisen kan worden voldaan. In plaats van deelsystemen kunnen we ook spreken van conceptuele componenten (SELECT Perspective spreekt van service packages, zie [Allen]).

Bij een componentgebaseerde architectuur kies je deelsystemen op zo'n manier dat redundantie in ge-

vens en processen wordt voorkomen. En je probeert om kennis over een bepaald onderwerp zoveel mogelijk binnen één deelsysteem te concentreren. Zo kunnen componenten gecreëerd worden die eenvoudig te hergebruiken, te wijzigen of te vervangen zijn.

De informatiesysteem-architect houdt zich dus vooral bezig met logische ordening. Hij scheidt proces-specifieke functionaliteit van procesonafhankelijke functionaliteit en clustert deze vervolgens tot (deel)systemen of conceptuele componenten. Elke component heeft een hoge interne samenhang en onderling hebben componenten een zo laag mogelijke koppeling. Het eindproduct vormt enerzijds de basis voor de ontwikkeling van de software-architectuur. Anderzijds is het nodig voor de ontwikkeling van het migratieplan.

3. SOFTWARE-ARCHITECTUUR De software-architectuur moet een beeld geven van de eisen die aan de software worden gesteld en beschrijft de technische oplossingen die zijn gekozen om aan die eisen te voldoen. Concrete richtlijnen beschrijven hoe de software tijdens de bouw moet worden gestructureerd.

Zo leggen we vast hoe conceptuele componenten (uit de informatiesysteem-architectuur) verder opgesplitst moeten worden tot software-componenten, hoe deze software-componenten onderling communiceren en hoe ze over de technische infrastructuur moeten worden verdeeld. Voorbeelden van veel gekozen oplossingen op software-architectuurniveau zijn de lagen architectuurstijl (layer pattern) en het broker pattern (zie [Buschmann]).

De software-architect selecteert dus maatregelen die invulling moeten geven aan de eisen van de business. Maar ook het specifiek invullen van zo'n maatregel hoort tot zijn werk. Bijvoorbeeld bij een niet-functionele eis als portabiliteit, moet niet alleen de vraag beantwoord worden of het layer pattern een goede oplossing biedt voor deze business-eis. Ook moet uitgezocht worden welke lagen nodig zijn, wat iedere laag precies mag bevatten en welke regels gelden bij lagen die elkaars componenten gaan aanroepen.

TYPEN ARCHITECTUURPRODUCTEN Nu we de architectuurniveaus hebben beschreven, gaan we kijken naar de architectuurproducten. Ondanks een veelheid aan mogelijke producten, onderscheiden wij een beperkt aantal typen architectuurproducten, die op de verschillende niveaus de kern van een ICT-architectuur kunnen weergeven.

1. EISEN EN MAATREGELN De ICT-architectuur van de ene organisatie kan terecht sterk verschillen van die van een andere, gelijksoortige organisatie. Als de eisen vanuit de business anders zijn, zullen andere maatregelen genomen moeten worden. Architectuur begint dus met het inventariseren van de eisen van de business (zie [Tapscott] hoofdstuk 10 en [Van Zeist] hoofdstuk 2). Per eis moet vervolgens bepaald worden welke maatregelen op welke architectuurniveaus nodig zijn om aan de eis te kunnen voldoen. Afstemming met de opdrachtgever en deskundigen over mogelijkheden, consequenties en kosten is hierbij noodzakelijk.

Een aantal maatregelen staat zo centraal in het huidige architectuurdenken, dat ze als aparte producttypen zijn opgenomen: de indeling in deelsystemen (zie punt 2) en de indeling in lagen (zie punt 3). Deze maatregelen bepalen zo sterk de structuur van het systeem, dat ze ook wel architectuurstijlen worden genoemd.

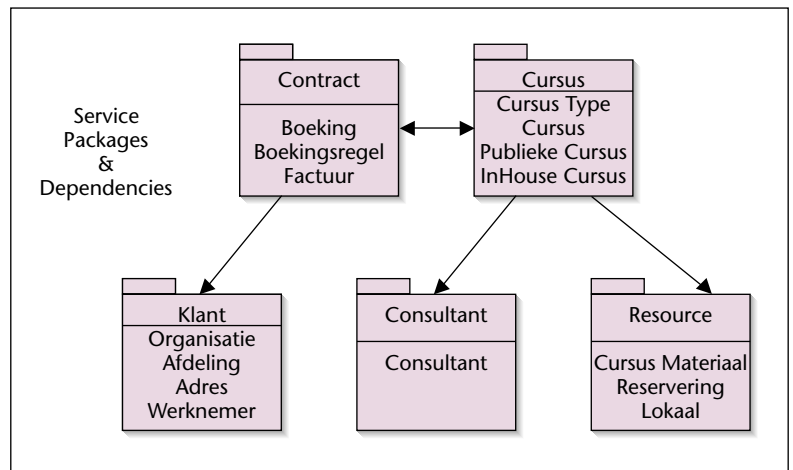
2. INDELING IN DEELSYSTEMEN Eén van de kernproblemen bij architectuur is hoe een systeem zo geordend kan worden in deelsystemen, dat aan de business-eisen voldaan wordt (zie figuur 1). Spilt de business-eis zich bijvoorbeeld toe op "snelheid en prijs van verwerking" dan verdienen procesgerichte ordeningscriteria de voorkeur. Als de eis "flexibiliteit" is, dan verdienen functionele (denk aan componentgebaseerde) ordeningscriteria de voorkeur.

Een systeem of deelsysteem hoeft hier niet de betekenis van ICT-systeem te hebben. Op businessniveau gaat het bijvoorbeeld om het inventariseren van de bedrijfsdomeinen of bedrijfsfuncties, hun onderlinge relaties en hun toewijzing aan de organisatiestructuur. Denk hierbij aan de discussie rondom e-business en de structuur van netwerkorganisaties.

Een architectuurproduct van dit type moet duidelijk maken:

- Welke deelsystemen er zijn.
- Wat het opdelingscriterium is.
- Wat de verantwoordelijkheid is van ieder deelsysteem, en welke onderdelen binnen het systeem vallen.
- Welke diensten (of interfaces) het deelsysteem aanbiedt.
- Wie in de organisatie verantwoordelijk is voor welk deelsysteem.

3. INDELING IN LAGEN Gelaagde architecturen zijn sinds de ontwikkeling van het client/server-model zeer populair geworden. Een zuiver lagenmodel staat alleen interactie tussen twee aangrenzende lagen toe en geen interactie over lagen heen (zie figuur 2). Door een goede



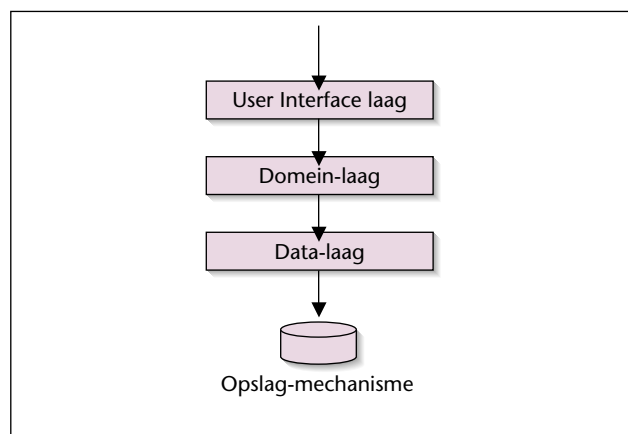
FIGUUR 1. Functionele componenten voor een cursusadministratie

opzet en toepassing van een lagenmodel kan hergebruik van functionaliteit in de onderste lagen bereikt worden en kunnen wijzigingen in technologie vaak beperkt worden tot wijzigingen in één laag.

Lagenmodellen worden vaak op het niveau van de software-architectuur toegepast, maar kunnen, bijvoorbeeld gericht op hergebruik, ook op de niveaus van business- en informatiesysteem-architectuur worden toegepast.

4. DEPLOYMENTMODEL Op ieder architectuurniveau is het belangrijk om te bepalen welke middelen nodig zijn en wat die voor een gekozen oplossingsrichting moeten doen. Belangrijk onderdeel vormen natuurlijk de typische ICT-middelen. De keuze voor ICT-middelen en de manier waarop deze belast worden, moet je op het niveau van de software-architectuur maken. Denk hierbij aan keuzes voor hardware, netwerken, systeemsoftware en middleware (zie figuur 3).

Ook op de niveaus van de business- en informatiesysteem-architectuur kan deze problematiek overigens niet onbesproken blijven, omdat haalbaarheid en kosten de keuzes ook hier beperken.



FIGUUR 2. Technisch lagenmodel

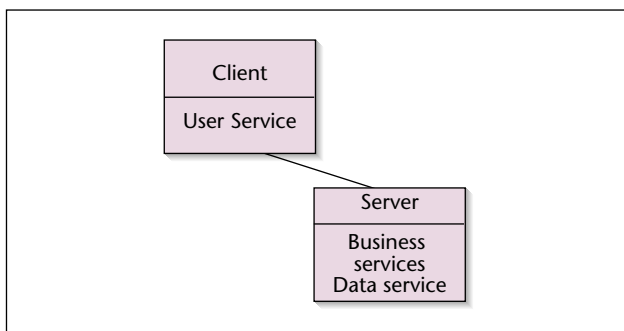
Product Architectuur	Eisen & maatregelen	Indeling in deelsystemen	Indeling in lagen	Deployment-model	Migratieplan
Business-architectuur	Business-eisen, principes en kwaliteitseisen	Business domeinen en/of bedrijfsfuncties	Herbruikbaarheid van bedrijfsfuncties	Verdeling van werk over mensen en/of machines	Volgorde van veranderingen in de organisatie
Informatie-systeem-architectuur	Functionele maatregelen (b.v. een standaard analyse-methode)	Functionele componenten	Herbruikbaarheid van informatiefuncties	Verdeling van functionaliteit over functionele componenten	Volgorde van veranderingen in de informatie-systemen
Software-architectuur	Technische maatregelen (b.v. ontwerpen programmeerrichtlijnen)	Technische componenten	Herbruikbaarheid van services	Verdeling van software componenten over hardware	Volgorde van veranderingen in de software componenten

TABEL 1. Architectuurproducten per architectuurniveau

5. MIGRATIEPLAN Het is belangrijk om een “ideale” architectuur op te stellen, zonder rekening te houden met de bestaande situatie en de benodigde migratie-inspanning. Dat ideale beeld moet ook gecommuniceerd worden, omdat het leidend kan zijn bij eventuele oplossingsalternatieven. Maar grote veranderingen kunnen het best stapsgewijs worden doorgevoerd. Een plateauplanning (zie Klinkenberg) kan ervoor zorgen dat het architectuurproces niet door al te grote ambities uit- of vastloopt.

PRODUCTEN PER ARCHITECTUURNIVEAU In de vorige paragrafen hebben we eerst uiteengezet op welke niveaus je architectuur kunt bedrijven. Daarna hebben we besproken welke producten kenmerkend zijn voor architectuur binnen de ICT. Deze twee zaken combineren we nu door de architectuurniveaus en -producten in een raamwerk tegen elkaar uit te zetten (zie tabel 1). Verdere invulling van dit raamwerk geeft meer duidelijkheid over de kern van de verschillende producten op de genoemde architectuurniveaus.

Wij geven hier een globale invulling, maar iedere organisatie die zich bezig houdt met architectuur zou het raamwerk verder in moeten vullen, aangepast aan de eigen doelen, middelen en methoden. Een gedetail-



FIGUUR 3. Deploymentmodel

leerde invulling van ieder “vakje” binnen het raamwerk maakt architectuur binnen de ICT grijpbaar, werkbaar, toetsbaar en overdraagbaar. Iedere cel in tabel 1 kan beschreven worden door het doel van het product, de benodigde werkwijze, de manier waarop het product genoteerd kan worden (diagrammen en/of tekst) en een opsomming van de achterliggende architectuurprincipes.

Literatuur

- Paul Allen, *Component-Based Development for Enterprise Systems (Applying the SELECT Perspective)*, Cambridge 1998.
- Frank Buschmann, *Pattern Oriented Software Architecture - a System of Patterns*, Wiley, 1996.
- IEEE 1471-2000, *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*, IEEE Computer Society, 2000.
- Niels Klinkenberg, *Plateauplanning*; Architectuur & Infrastructuur nummer 1, 2000
- Don Tapscott, *Paradigm shift*, McGraw-Hill, 1993.
- John Zachman, *A framework for information system architecture*; IBM Systems Journal, Vol 26, No 3, 1987; zie ook <http://www.zifa.com/>.
- Bob van Zeist, *Kwaliteit van softwareproducten*; Kluwer 1996.

ing. Joost A.P. Lommers werkt bij ISES International als docent/consultant op het gebied van architectuur, ontwikkelscenario's en moderne systeemontwikkeling

drs. Leo. J. Pruijt is als docent verbonden aan de Hogeschool van Utrecht, afdeling ICM, opleiding Informatica