

Competenties van de ICT-Architect

Roel Wieringa

Competenties van de ICT-Architect

Roel Wieringa

Universiteit Twente

<http://is.cs.utwente.nl/~roelw>

roelw@cs.utwente.nl

Zondag 6 November 2005

1.	Inleiding	2
2.	Competenties en beheersingsniveau's	2
3.	What's in a Name?	3
4.	Wat is een architect?	4
5.	De competentie-ijsberg	5
6.	Attitudes en intermediaire vaardigheden	6
6.1.	Persoonlijkheidskenmerken	6
6.2.	Normen en waarden	8
6.3.	Intermediaire vaardigheden	8
7.	Vaktechnische competenties	9
8.	Rollen van een architect.....	12
9.	Discussie	15
10.	Literatuur.....	16

1. Inleiding

In de afgelopen twee jaar is door de Universiteit Twente in opdracht van het Nederlands Architectuurforum (NAF) onderzoek gedaan naar competenties van ICT-architecten.¹ In dit artikel analyseer ik de resultaten van dit onderzoek. Deze analyse doe ik op persoonlijke titel en hij vertegenwoordigt dus niet het standpunt van het NAF.

2. Competenties en beheersingsniveau's

De volgende definities maakt duidelijk dat competenties gebaseerd zijn op kennis, vaardigheden en attitudes.

- Een **competentie** is een vermogen om in een concrete situatie adequaat te handelen, gebaseerd op kennis, attitude, en vaardigheden [DV2003].

Bijvoorbeeld, de competentie om auto te kunnen rijden bestaat uit het vermogen om in concrete verkeerssituaties adequaat als bestuurder van een auto te reageren, gebaseerd op kennis van de verkeersregels, de vaardigheid om te sturen, en de gewenste attitude van een verkeersdeelnemer. Een belangrijk deel van deze kennis, vaardigheid en attitude zijn tijdens rijlessen geleerd; een ander deel is daarna pas geleerd, en een deel van de attitudes, zoals respect voor andere verkeersdeelnemers, komt voort uit persoonlijkheidskenmerken die al aanwezig waren voordat de autorijder rijlessen ging volgen. Tijdens het rij-examen wordt de competentie om rijvaardigheden, verkeersregels en verkeersattitudes in concrete situatie toe te passen, getoetst.

Een competentie kan op allerlei niveau's beheerst worden. Het blijkt nuttig te zijn om dit in twee dimensies te classificeren, *ervaring* en *inzicht*. De meeste competentie-modellen onderscheiden drie ervaringsniveau's:

1. **Geleerd.** De competentie is geleerd, maar nog niet gebruikt. De student heeft de theorie bestudeert en het theorie-examen met goed gevolg afgelegd.
2. **Toegepast.** De competentie is onder begeleiding toegepast. Niet alleen beheerst de student de theorie, hij heeft hem ook in een practicum toegepast.

¹ In Januari 2004 is onder leiding van Erik Proper (Radboud Universiteit Nijmegen) en Pascal van Eck (Universiteit Twente) een Brown Paper sessie met NAF leden gehouden om competenties van ICT-architecten te inventariseren. Vervolgens is door Koen Voermans (Universiteit Twente) en Pascal van Eck een analyse van de competenties van ICT-architecten volgens het NGI model van Op de Coul [ODC2001]. In de herfst van 2004 hebben Koen Voermans en Claudia Steghuis (UT) de deelnemers van LAC 2004 geënquêteerd naar hun visie op gewenste competenties van ICT-architecten [VSW2004]. De resultaten hiervan zijn gecombineerd met de resultaten van eerder onderzoek in een voorlopig rapport van de Werkgroep Professionalisering van de Architect van het NAF [WV2005]. Vervolgens zijn deze resultaten voor Claudia Steghuis en Koen Voermans in een breder competentiekader gezet [SVW2005a] en tenslotte zijn leden van de Werkgroep Professionalisering geïnterviewd naar hun visie op ICT-architectcompetenties [SVW2005b]. Al deze verslagen zijn interne NAF-rapporten; de huidige analyse is openbaar en is bedoeld als impuls voor verdere discussie in bredere kring.

3. **Gebruikt.** De competentie is zelfstandig toegepast. Dat betekent dat de eigenaar van de competentie in een concrete situatie adequaat kan handelen, gebaseerd op kennis, vaardigheden en attitude. Dit beheersingsniveau kan naar wens onderverdeeld worden in ervaringsjaren: Minstens 3 jaar ervaring, meer dan 5 jaar ervaring, etc.

Een tweede classificatie van competenties is naar inzicht.

1. **Beheersing.** Men kan adequaat handelen, maar niet rechtvaardigen waarom nu juist deze actie is gepleegd en geen andere. Met ander woorden, men kan geen alternatieven aangeven of voorkeursvolgordes van alternatieven aangeven.
2. **Verantwoording.** Men kan, voordat de actie in een concrete situatie gepleegd wordt, aangeven waarom deze actie de voorkeur heeft boven anderen.
3. **Reflectie.** Men kan niet alleen de actie verantwoorden, maar ook achteraf die in een breder kader en zelfs ander kader plaatsen, waarbij ook het eigen functioneren betrokken wordt. Men kan bijvoorbeeld aangeven dat de actie gepleegd werd vanuit optiek dat het om een informatievoorzieningsprobleem ging, terwijl achteraf bleek dat het veel meer om een procescoördinatieprobleem ging, en hieruit lessen voor het eigen toekomstige handelen trekken. Reflectie houdt dus in probleemkaders kunnen wijzigen, het eigen functioneren kunnen evalueren, en permanent leren de eigen competenties te verbeteren.

Combineren we de twee dimensies, dan ontstaat de volgende matrix.

		Inzicht		
		Beheersing	Verantwoording	Reflectie
Ervaring	Geleerd	Conceptueel		
	Onder begeleiding toegepast	Toegepast		
	Zelfstandig gebruikt.	Zelfstandig	Gevorderd	Expert

De meeste bedrijven onderscheiden ongeveer vijf beheersingsniveau's van competenties, die ongeveer zoals in de voorgaande tabel ingevuld worden. Als een student een opleiding verlaat heeft hij of zij conceptuele en toegepaste competenties. Die kunnen, afhankelijk van de kwaliteiten van de student op verantwoordingsniveau of zelfs reflectieniveau uitgeoefend worden. Zonder zelfstandige ervaring is het niet relevant deze niveau's verder te onderscheiden.

Na enige jaren zelfstandige beroepsuitoefening is het competentieniveau Zelfstandig. Afhankelijk van de kwaliteit van de architect kan de architect een Gevorderd of Expert-niveau van competentie hebben.

3. What's in a Name?

Voordat ik me waag aan het benoemen van architectcompetenties, wil ik wat relativerende opmerkingen maken. Sommige discussies in Architectuurland gaan over het woordenboek, en niet over architectcompetenties. Is functie X nu een *Business Architect* of een *Enterprise Architect*? Dergelijke discussies kunnen soms met hoog oplopende emoties gevoerd worden, waarbij de emoties hoger te lijken worden

naarmate de termen minder gedefinieerd zijn. Terwijl het belang hiervan is in vele gevallen niet meer is dan de vraag of iets nu een fiets of een tweewieler moet heten.

Achter de woordenboekdiscussies is wel enig commercieel belang te ontdekken: De partij die zijn terminologie tot standaard weet te verheffen, heeft een commercieel voordeel op een partij die zijn terminologie, en dus ook zijn competentiegebouw, moet aanpassen aan de standaard.

Bovendien is de betekenis van woorden die in een standaard gedefinieerd zijn, niet meer een zaak van pure conventie. Is een surfplank een vaartuig? Met die vraag zijn geen commerciële belangen gemoeid, maar toch de vraag is meer dan een discussie onder lexicografen die een woordenboek aan het schrijven zijn: Als een surfplank een vaartuig zou zijn, dan zegt ons wetboek dat de bestuurder in het bezit moet zijn van een vaarbewijs. In Nederland heeft de rechter bepaald dat een surfplank geen vaartuig is. En dus hoeven surfers in Nederland geen vaarbewijs te hebben. Naar analogie hiermee, als er een standaard voor de competenties van ICT-architecten komt, dan is de vraag of een software-ontwerper een ICT-architect is niet meer een zaak van lexicografie maar van personeelsmanagement. De standaard heeft dan namelijk implicaties voor de competenties die van een software-ontwerper gevraagd worden.

Het feit dat ik in dit artikel bepaalde architectnamen gebruik heeft geen enkele normatieve consequentie. Ik gebruik voor elke architectrol de meest voorkomende naam, maar voor elke rolnaam is er minstens één bedrijf te vinden die een andere naam gebruikt.

4. Wat is een architect?

Besprekingen in de literatuur over wat een architect is, en wat hem of haar van een ontwerper of ingenieur onderscheidt, ontaarden nog wel eens in zweverige verhalen over schoonheid (Christofer Alexander) en klassieke architectuur (Vitruvius). In de praktijk blijkt er consensus in het gebruik van het woord “architect” te zijn, die een stuk minder zweverig is. Centraal hierin staat de volgende taak van de architect:

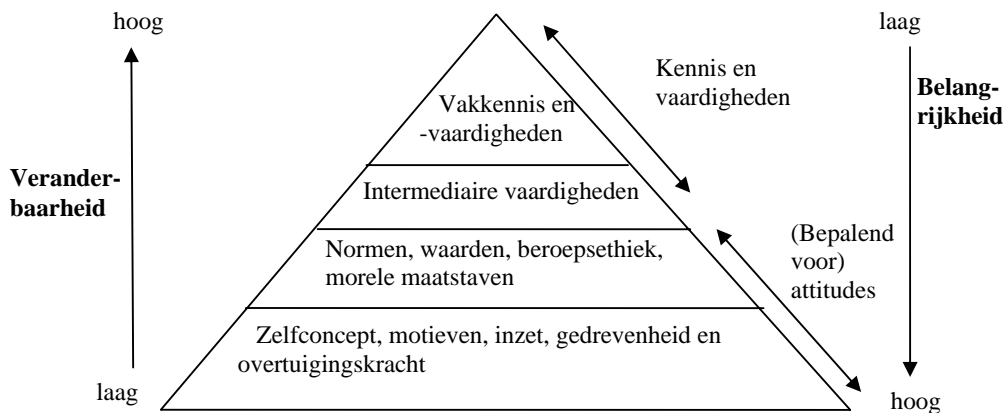
- Een ICT-architect analyseert bedrijfsbehoeften en ontwerpt daarbij passende ICT-oplossingen.

Dit is analoog aan de taak van een fysieke architect: Die analyseert de behoeften van de klant en ontwerpt een bijpassende fysieke structuur die aan die behoeften tegemoet komt. Een architect moet dus in staat zijn om met de klant mee te denken, zijn taal te spreken, zijn doelen te begrijpen, en zijn problemen te analyseren. Daarnaast moet hij ook op de hoogte zijn van de huidige stand van de oplossingstechnieken (bouwmaterialen voor de fysieke architect, softwarecomponenten voor de ICT-architect) en in staat zijn om die zodanig te combineren dat een oplossing voor de behoeften van de klant ontstaat. De architect moet dus ook de taal van de technologen kunnen spreken, en met de bouwer kunnen praten over de manier waarop de oplossing geconstrueerd zal worden. En dit vat al een eerste versie van de gewenste competenties van een architect samen: Met opdrachtgever en uitvoerder kunnen praten, oplossingen kunnen ontwerpen, meerwaarde kunnen creëren. Maar daarover zodadelijk meer.

Architecten bemoeien zich niet met de details van de oplossing maar met de grote lijn, en meer in het bijzonder met de manier waarop de oplossingscomponenten gezamenlijk een meerwaarde hebben voor de klant. Architecten hebben dus een competentie om *systeem* te zien, waarbij ik “systeem” gebruik in de betekenis van *synergie, samenhang met meerwaarde*. Architecten hebben het overzicht over de klantbehoeften en de oplossingscomponenten om in te zien op wat voor manier de samenhang van de oplossingscomponenten een meerwaarde creëert voor de klant.

5. De competentie-ijsberg

Architecten moeten zowel de taal van de klant als van de oplossingstechnoloog kunnen spreken. Het is opvallend dat vooral bij gebruikersbedrijven ICT-architecten zowel een achtergrond kunnen hebben in “de business” als in de ICT. Zo zijn er ICT-architecten die een achtergrond als verzekeringsspecialist of werktuigbouwkundige hebben, naast architecten die een achtergrond als software engineer hebben. De belangrijkste competentie van ICT-architecten is blijkbaar niet zozeer vaktechnisch — dat is aan te leren— maar communicatief. Dit kan aan de hand van de “ijsbergstructuur” van Bergenhenegouwen *et al.* [BMT1999] verduidelijkt worden.



De bovenste laag bevat **vaktechnische kennis en –vaardigheden**. Dit zijn waarneembare kennis en vaardigheden die voor de uitoefening van een vak nodig zijn. Een ICT architect moet bijvoorbeeld kennis hebben van het bedrijf en van ICT. Ik zal deze competenties later samenvatten. Competenties in de top van de ijsberg zijn waarneembaar en ze zijn goed onderwijsbaar. Alle competenties daaronder zijn niet goed waarneembaar en niet goed onderwijsbaar. Helaas zijn die belangrijker voor de uitoefening van competenties dan de kennis en vaardigheden in het topje van de ijsberg.

Intermediaire competenties zijn van belang voor de uitoefening van meerdere vakken. ICT architecten, werktuigbouwkundigen, verzekeringsexperts en bankiers doen allen in meer of mindere mate een beroep op deze competenties. Dit zijn **breed inzetbare beroepscompetenties** zoals

- sociale en communicatieve vaardigheden
- algemeen technische en beroepsmatige inzichten

- organisatorische kwaliteiten
- handelingsoriëntaties

Onder de intermediaire competenties liggen **normatieve attitudes** over “hoe wij dingen moeten doen”. Hier hoort ook een beroepsethiek bij, maar ook algemene morele normen en waarden horen in deze laag. Competenties op dit niveau zijn uiterst moeilijk aan te leren en hebben alles te maken met het socialiseren in een groep, iets wat bij uitstek een langdurig en zeer persoonlijk proces is, waarin de identiteit van een persoon als lid van een groep gevormd wordt.

De onderste laag competenties bevat **persoonlijkheidskenmerken** die niet goed onderwijsbaar zijn. Persoonlijkheidskenmerken kunnen evolueren en doen dat altijd in interactie met de omgeving. Het zijn echter geen kenmerken die door een cursus aangeleerd kunnen worden. Cursussen in bijvoorbeeld leiderschap, een van de persoonskenmerken, zijn er op gericht de cursisten zich bewust te maken van hun leiderschapskwaliteiten en ze te leren die kwaliteiten verder te ontwikkelen. Dat is een ander cursusdoel dan iemand de nieuwste ontwikkelingen rond service-oriented architecturen te leren. Cursussen die op persoonlijkheidskenmerken gericht zijn, doen een appèl op de persoonlijkheid van elke cursist.

Het woord “competenties” wordt in de ijsberg niet gebruikt. Dat komt omdat competenties uitgeoefend worden *op basis van* kennis, vaardigheden en attitudes, en die komen wel in de ijsberg voor. Kennis en vaardigheden komen in de bovenste twee lagen voor. Attitudes worden niet genoemd, maar volgen uit de normen, waarden en persoonlijkheidskenmerken in de onderste twee lagen.

Bij de selectie van personen voor ICT-architecten zijn de vaktechnische competenties van ondergeschikt belang. Het blijkt dat als iemand de juiste persoonlijke kenmerken, normen en waarden, en intermediaire competenties heeft, hij of zij zich de benodigde ICT-competenties en bedrijfs-gerelateerde competenties in de bovenste laag in voldoende mate aan kan leren. Ik bespreek dus nu eerst de competenties en attitudes in de onderste drie lagen en daarna pas de vaktechnische competenties van de ICT-architect.

6. Attitudes en intermediaire vaardigheden

6.1. Persoonlijkheidskenmerken

De “competenties” die we in de verschillende sessies, enquêtes en interviews verzameld hebben, bevatten een groot aantal persoonlijkheidskenmerken. Dat zijn dus feitelijk geen competenties, maar ze bepalen wel attitudes waarmee competenties uitgeoefend worden. Psychologen hebben persoonlijkheidskenmerken in vijf groepen verdeeld, die door Goldberg nog eens nader gesystematiseerd zijn [G1990]. Ieder persoon kan gekarakteriseerd worden door kenmerken uit die vijf groepen te kiezen. Wij hebben de volgende persoonlijkheidskenmerken van ICT-architecten verzameld. Achter elk kenmerk staat het aantal verschillende bronnen dat die competentie genoemd heeft.

1. **Extravertie.**
 - Communicatief (10)
 - Initiatiefrijk (4)
 - Bereid uitdagingen aan te gaan (1)
 - Kunnen overtuigen (1)
2. **Aangenaamheid (Agreeableness).**
 - Teamplayer (7)
 - Empatisch (andermans standpunt kunnen innemen) (4)
 - Luistervaardig (4)
 - Vertrouwenwekkend (1)
3. **Betrouwbaarheid (Dependability, conscientiousness).**
 - Analytisch (7)
 - Georganiseerd, systematisch en ordelijk (7)
 - Besluitvaardig (2)
 - Resultaat/doelgericht (2)
 - Integer (1)
 - Accuraat (1)
 - Doorzettingsvermogen (2)
4. **Emotionele stabiliteit**
 - Zelfstandig (3)
 - Stressbestendig (1)
5. **Intellect (reflectie)**
 - Creatief /innovatief (8)
 - Abstractievermogen (5)
 - Zichzelf blijven ontwikkelen (5)

Kijken we alleen naar de meest frequent genoemde kenmerken, dan blijkt dat een architect hoog moet scoren op reflectie (genoemd worden creativiteit, abstractievermogen en blijven leren) en vervolgens ook op extravertie (communicatief ingesteld), een teamplayer die goed naar anderen luistert, een betrouwbaar persoon die problemen en oplossingen systematisch analyseert.

Ik ken niemand die al deze eigenschappen heeft. Er zijn uit deze lijst enkele tegenstrijdige karikaturen van architecten te construeren. Hier zijn er twee:

- *De masculiene architect*: resultaatgericht, besluitvaardig, met overtuigingskracht;
- *De feminiene architect*: een sensitieve teamplayer die goed luistert.

Twee andere karikaturen zijn die van de artiest versus de analyticus:

- *Artiest*: Iemand die op hoog abstractieniveau creatieve oplossingen voorstelt;
- *Analyticus*: Iemand die nauwgezet en systematisch problemen analyseert en oplossingen uitwerkt.

Blijkbaar zijn er verschillende types personen die goed als architect kunnen functioneren.

6.2. Normen en waarden

Van alle door ons verzamelde competenties zijn er twee die in feite waarden zijn (en dus geen competenties). Dat zijn

- organisatie-gerichtheid (7)
- klantgerichtheid (8)

Hiermee bedoelen we niet *kennis* van de organisatie of klant, hoewel die natuurlijk ook aanwezig moet zijn, maar de *bereidheid* de waarden van de organisatie en van de klant mee te nemen in adviezen over architectuur. Net zoals persoonlijkheidskenmerken bepaalt dit in belangrijke mate de attitude waarmee competenties in concrete situaties worden uitgeoefend.

6.3. Intermediaire vaardigheden

Architecten moeten volgens de door ons onderzochte bronnen een aantal intermediaire vaardigheden hebben, die voor de uitoefening van allerlei beroepen van belang zijn, in het bijzonder dus voor de uitoefening van het beroep van ICT-architect. De intermediaire vaardigheden in zijn te groeperen onder de kopjes Managementvaardigheden, Consultancy-vaardigheden, en Ontwerpvaardigheden.

- **Management-vaardigheden**
 - Schriftelijke en mondelinge communicatie (10)
 - Leiderschap (9)
 - Projectmanagement (2)
 - Portfoliomanagement (1)
 - Programmamanagement (2).
 - Verandermanagement (1)
- **Consultancyvaardigheden**
 - Onderhandelvaardigheden (4)
 - Consultancyvaardigheden (3)
 - Didactische vaardigheden (3)
- **Ontwerpvaardigheden**
 - Oplossingen ontwerpen (1)
 - Requirements engineering (Oplossingen op problemen afstemmen) (1)
 - Architectuurprincipes toepassen (1)
 - Sourcing (1)
 - Requirements management (1)
 - Kosten/batenanalyse van oplossingen doen (1)

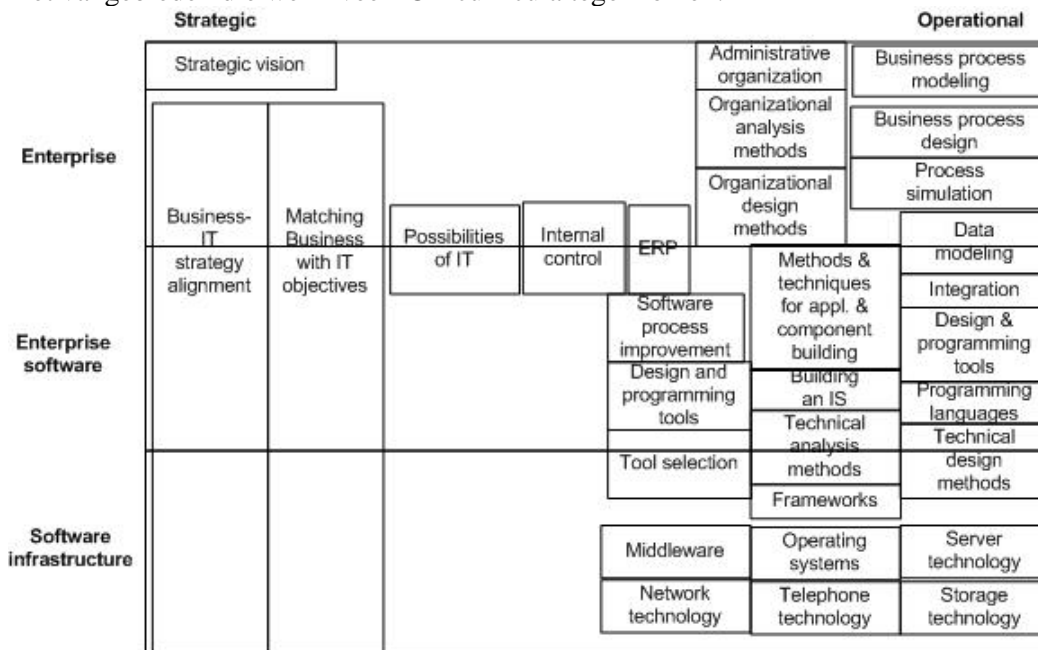
Als eerste springt naar voren dat een architect leiderschaps- en communicatieve vaardigheden moet hebben. Echter, dit hangt m.i. samen met de rol die een architect moet spelen. Als een architect business unit managers en projectmanagers moet overtuigen van de waarde van een door hem ontworpen architectuur, dan zijn leiderschap en communicatie nuttige vaardigheden. Als de architect dat niet hoeft te doen, dan zijn dat minder nuttige vaardigheden.

Meer in het algemeen zijn er verschillende meningen over de mate waarin architecten de genoemde management- en consultancyvaardigheden moeten bezitten. Dit kan m.i. alleen uitgewerkt worden als we meer weten over de rollen die architecten in organisaties spelen.

Het lijstje met ontwerpvaardigheden is heel anders. Dat zijn vaardigheden die ieder, die oplossingen voor concrete problemen moet verzinnen, moet hebben: Een architect moet oplossingen verzinnen (ontwerpen), afstemming tussen oplossingen en problemen controleren, architectuurprincipes kunnen toepassen, beslissen waar de oplossingselementen vandaan komen (sourcing), alle eisen aan de oplossing managen, en een schatting van de financiële consequenties van een oplossing kunnen maken. Opvallend is dat *Probleemonderzoek* niet bij het lijstje van vaardigheden staat, terwijl dit toch een taak is die architecten altijd moeten uitvoeren om een goede oplossing te bedenken.

7. Vaktechnische competenties

Het topje van de ijsberg van architectcompetenties bestaat uit vaktechnische kennis en vaardigheden. Het volgende diagram geeft de door onze bronnen genoemde vaardigheden weer. Links staan de strategische kennis en vaardigheden, rechts de operationele. De verzameling competenties beslaat een min of meer vertrouwd lijst met vakgebieden die we in veel ICT-curricula tegenkomen.



In dit plaatje bedoel ik met de **software-infrastructuur** alle software die altijd voor iedereen beschikbaar moet zijn: Bedrijven verdelen de software-infrastructuur meestal in infrastructuurdomeinen, zoals operating systems, middleware, file servers, web servers, telefonie, etc. Elk domein is een kennisgebied en correspondeert met een vaktechnische competentie. Welke vaktechnische infrastructuurcompetenties precies

erkend worden, verschilt per bedrijf. Toch is er een hoge mate van uniformiteit, die in voorgaand plaatje uitgedrukt wordt.

Enterprise software bestaat uit de informatiesystemen en applicaties die bepaalde bedrijfsprocessen ondersteunen. In tegenstelling tot infrastructuur, is enterprise software alleen voor sommige gebruikers beschikbaar, en worden beschikbaarheidseisen afgeleid uit de specifieke processen waarvoor de enterprise software ondersteuning moet bieden. Bedrijven onderscheiden hier een aantal competenties die alle op het gebied van software engineering en information systems engineering liggen.

Tenslotte is er een aantal **bedrijfsgerichte** competenties, zoals administratieve organisatie, procesmodellering en interne controle.

Al deze competenties worden op operationeel niveau uitgeoefend. Op het gebied van de infrastructuur zijn de competenties gericht op technologiedomeinen, daarboven zijn de competenties gericht op ontwerpmethoden: Softwareontwerp voor enterprise systems resp. organisatie-ontwerp voor enterprises.

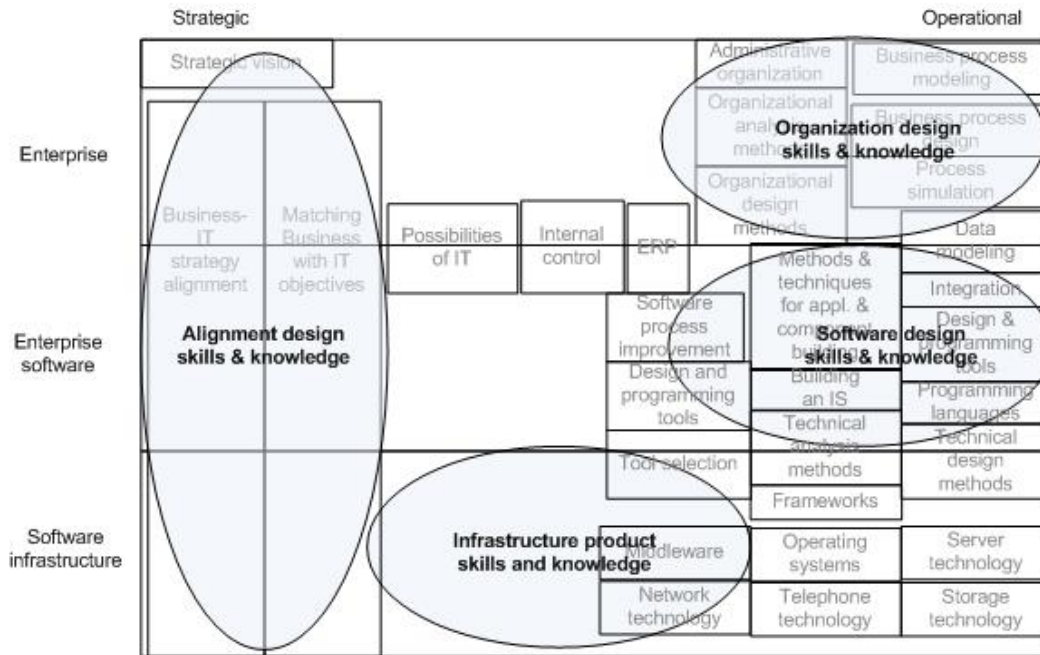
De figuur noemt ook drie strategische vaardigheden:

- Strategic vision
- Business-IT strategy alignment
- Matching business with IT objectives

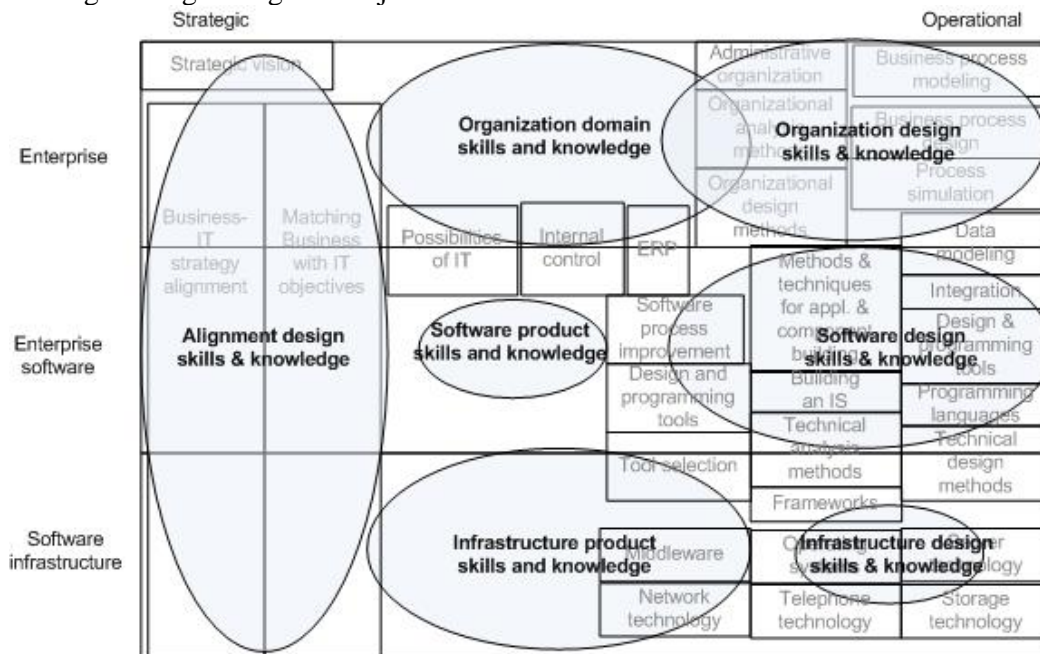
Dit zijn niet zulke vakspecifieke vaardigheden en ze hadden ook bij intermediaire vaardigheden genoemd kunnen worden. Ze gaan alle drie op het afstemmen van ICT op organisatiedoelen, op strategisch niveau. Waar requirements engineering, genoemd als intermediaire vaardigheid, vooral op individuele oplossingen gericht is, zijn deze strategische vaardigheden vooral op het bedrijf en zijn ICT assets als geheel gericht. Het is voor te stellen dat deze vaardigheid in twee varianten komt:

- Strategische afstemming van infrastructuur op bedrijfsdoelen
- Strategische afstemming van alle enterprise systems op bedrijfsdoelen.

Dit onderscheid wordt door onze bronnen echter niet gemaakt.



De vaktechnische competenties kunnen in drie gebieden gegroepeerd worden, zoals in de bovenstaande figuur aangegeven is. Op enterprise niveau is er vooral sprake van ontwerp-kennis en -vaardigheden, zoals bedrijfsprocesmodellering en -ontwerp, organisatie-ontwerp, etc. Ook op enterprise software-niveau is er vooral sprake van ontwerp-kennis, zoals methoden en technieken voor software-ontwerp, programmeertalen en software-frameworks. Op infrastructuurniveau is er daarentegen vooral sprake van productkennis: operating systems, middleware, servetechnologie etc. De reden voor dat verschil is snel te vinden: Infrastructuur-software wordt gekocht, niet gebouwd. Enterprise-software wordt daarentegen blijkbaar vaker gebouwd dan gekocht. De voorgaande figuur laat een paar witte vlekken zien, die in de volgende figuur ingevuld zijn.



We hebben nu twee typen vaktechnische competenties.

- **Domeinkennis en –vaardigheden.**
 - Op organisatorisch niveau is dit kennis van het bedrijfsdomein. Bijvoorbeeld in de financiële wereld verzekeringen, hypotheeken, beleggen etc. Het belang van organisatiedomeinkennis blijkt uit het feit dat architecten op dit niveau vaak “uit de business” komen en zich op het gebied van ICT bijgeschoold hebben, of uit de ICT komen en zich grondig in het bedrijfsdomein hebben verdiept.
 - Op enterprise softwareniveau gaat het om kennis van de markt van relevante softwareproducten. Naarmate enterprise software commodities worden, is deze markt groter.
 - Op infrastructuurniveau zijn de softwareproducten al commodities en architecten op dit niveau moeten deze markt van binnen en buiten kennen.
- **Ontwerpkennis- en vaardigheden.**
 - Aan de operationele kant van het plaatje moeten architecten op de hoogte zijn van de relevante ontwerpmethoden en –technieken. Infrastructuurarchitecten programmeren weinig, maar ook het invoeren en in gebruik nemen van infrastructuurproducten vergt het nodige aan ontwerpkennis.
 - Architecten aan de strategische kant van de figuur moeten bedrijfsstrategie, enterprise software strategie en infrastructuur strategie op elkaar afstemmen. Bovendien moeten ze het operationele ontwerp van software-oplossingen op dit strategische niveau kunnen afstemmen. In de praktijk wordt enterprise software alignment onderscheiden van infrastructuur alignment. Dat creëert ook een intern afstemmingsprobleem tussen deze twee soorten software.

Tot zover de analyse van vaktechnische competenties. Als laatste noem ik drie competenties die onze bronnen vermelden, maar die niet in de lagenstructuur passen:

- Security management
- Operational management
- Operational risk management

Het is de vraag of deze managementcompetenties tot de competenties van de architect horen. Waarschijnlijk bedoelden onze bronnen dat er architecten moeten zijn die dergelijke managementstructuren en –processen moeten kunnen ontwerpen.

8. Rollen van een architect

Een **rol** is een positie in een organisatie. Rollen worden meestal gedefinieerd door

- Verantwoordingslijnen te definiëren (wie rapporteert aan wie?)
- Diensten te definiëren (wat voor dienst moet de persoon die de rol speelt, leveren?)
- Taken te definiëren (wat moet de persoon die de rol speelt, doen om die diensten te leveren?)

In ons onderzoek hebben we ons tot het tweede aspect van rollen beperkt: Wat moet een rol bereiken? De andere twee aspecten moeten later nog onderzocht worden.

Elk bedrijf definieert andere rolstructuren, maar globaal genomen komen de volgende twee rolstructuren op veel plaatsen voor. In *gebruikersbedrijven* (banken, verzekeraars) onderscheidt men vaak de volgende drie rollen

- **Domeinarchitect.** Een architect die bedrijfsoplossingen bij bedrijfsproblemen en -doelen vindt. De oplossingen bestaan uit processen, structuren, informatiestromen etc. De domeinarchitect gaat niet in op de ICT-onderdelen daarvan. vaak is de domeinarchitect een domeinspecialist die talent heeft voor architectuurontwerp.
- **Enterprise-architect.** Een architect die enterprise systems ontwerpt om bedrijfsproblemen op te lossen of bedrijfsdoelen te bereiken, meestal op globaal niveau. Vaak worden de componenten van de oplossing op de markt gekocht en in geen geval ontwerpt de architect de interne structuur van een enkel softwaresysteem.
- **Infrastructuurarchitect.** Een architect die infrastructuur ontwerpt om bedrijfsproblemen op te lossen of bedrijfsdoelen te bereiken. De infrastructuren zijn kant-en-klare systemen die van derden gekocht worden.

Bij *aanbieders* (consultancybedrijven, vendors) is er een grotere variatie in architectrollen. Men onderscheidt vaak de volgende rollen:

- **Enterprise architect** zorgt voor afstemming enterprise software en het bedrijf
- **Informatie-architect** modeleert en ontwerpt informatiestromen
- **Procesarchitect** modeleert en ontwerpt bedrijfsprocessen
- **Applicatie-architect** ontwerpt applicatie-architecturen
- **Integratie-architect** ontwerpt de manier waarop verschillende enterprise systemen samen gaan werken.
- **Infrastructuur-architect** ontwerpt infrastructuur.

Het is wenselijk eenheid in deze terminologie te brengen, maar iedere aanbieder hecht veel waarde aan de rollen die hij zelf gedefinieerd heeft, en voelt er weinig voor om de rollenstructuur van een ander bedrijf over te nemen. Er is echter orde in deze veelheid waar te nemen. Het volgende plaatje laat verschillende aspecten van systemen zien vanaf de hardware infrastructuur tot aan de bedrijfsomgeving.

	Services	Behavior	Communication	Data	Quality for user	Quality for developer
Enterprise environment		<i>Customer behavior</i>	<i>Marketing channels</i>	<i>Market data</i>		
Enterprise	<i>Business services</i>	<i>Business processes</i>	<i>Information flows</i>	<i>Data semantics</i>		
Enterprise systems	<i>System functions</i>	<i>Dialog structure</i>	<i>Data flows</i>	<i>Logical data structure</i>		
Software infrastructure	<i>Transaction services</i>	<i>Protocols</i>	<i>Networking</i>	<i>Physical data structure</i>		
Physical infrastructure	<i>Software execution</i>	<i>Producing heat, noise</i>	<i>Cables</i>			

De verschillende kolommen van de figuur zijn systeemaspecten die op elk niveau in andere vorm terug komen en die over alle niveau's heen op elkaar afgestemd moeten worden. Als voorbeeld staat er in elke cel van de tabel een voorbeeld van het betreffende aspect op dat niveau. Er zijn vele andere voorbeelden te geven. De tabel is gebaseerd op uitgebreid onderzoek van raamwerken voor systems engineering, productontwerp en software engineering. We gebruiken hem om empirisch onderzoek naar architectuurafstemming te doen [RE2004].

De tabel is vooral op operationeel niveau bruikbaar. Op strategisch niveau is het onderscheid tussen de kolommen te gedetailleerd. Ontwerp op strategisch niveau zal op globaal niveau aan alle kolommen aandacht schenken. Kijken we nu eerst op operationeel niveau, dan is de volgende tabel een samenvatting van veel gebruikte terminologie.

	Services	Behavior	Communication	Data	Quality for user	Quality for developer
Enterprise environment						
Enterprise	<i>Business architect</i>	<i>Process architect</i>		<i>Informatie-architect</i>		
Enterprise systems	<i>Applicatie-architect</i>		<i>Integratie -</i>	<i>Data-ontwerper</i>		
Software infrastructure		<i>Infrastructuur-architect</i>				
Physical infrastructure						

In deze tabel is een *business architect* iemand die een deel van de organisatie ontwerpt, met name welke onderdelen er zijn en welk onderdeel welke diensten aan welk andere onderdeel levert. Dat is iets heel anders dan een *enterprise architect*, die in het nu geaccepteerde spraakgebruik iemand is die op strategisch niveau naar de afstemming kijkt van infrastructuur, softwaresystemen en enterprise, en daarbij naar alle aspecten van de tabel kijkt. Een enterprise architect is dus niet zomaar een hele dure business architect, hij of zij kijkt ook naar alle aspecten van business-ICT alignment. Gebruiken we het plaatje nu even op strategisch niveau, dan krijgen we dit:

	Services	Behavior	Communication	Data	Quality for user	Quality for developer
Enterprise environment						
Enterprise	<i>Enterprise architect</i>					
Enterprise systems						
Software infrastructure						
Physical infrastructure						

Een goede enterprise architect zal overigens ook de bedrijfsomgeving bij zijn of haar afstemmingsoverwegingen betrekken, en ook de fysieke infrastructuur niet vergeten.

Terug op de grond vinden we bij gebruikersbedrijven vaak een simpele invulling van de tabel:

	Services	Behavior	Communication	Data	Quality for user	Quality for developer
Enterprise environment						
Enterprise		<i>Domeinarchitect</i>				
Enterprise systems	<i>Enterprise architect / Applicatiearchitect</i>					
Software infrastructure		<i>Infrastructuurarchitect</i>				
Physical infrastructure						

In dit plaatje ontwerpt de domeinarchitect alle aspecten van bedrijfsoplossingen, en is hij/zij de natuurlijke gesprekspartner van de “enterprise architect” gebruikersstijl, die de volledige laag van enterprise software onder zijn/haar hoede heeft. de enterprise architect-gebruikersstijl praat weer met de applicatie-architect, die individuele applicaties ontwerpt binnen de oplossing die de enterprise architect-gebruikersstijl ontworpen heeft. de infrastructuurarchitect tenslotte ontwerpt de volledige infrastructuurlaag. Individuele infrastructuurproducten worden niet door gebruikers ontworpen maar gekocht.

De voorgaande plaatjes suggereren dat architecten vooral naar functionele eigenschappen kijken. Dit moet nader onderzocht worden. Mijn hoop is dat architecten op operationeel niveau in de praktijk juist ook naar kwaliteitsattributen kijken, zoals bruikbaarheid, performance, veiligheid, interoperabiliteit etc. het zijn vaak juist deze attributen die bepalen hoe waardevol een systeem voor een bedrijf is.

De vraag hoe we deze stapel van architectuurtaken in rollen verdelen, en hoe we elk van die rollen dan gaan noemen, is m.i. minder belangrijk dan de vraag of we elke architectuurtaak aan een rol hebben toegekend.

9. Discussie

In dit rapport heb ik een eerste overzicht gemaakt van de competenties van ICT architecten zoals onze bronnen die zien. De belangrijkste toevoeging aan de informatie die we van onze bronnen gekregen hebben, is het plaatsen van de competenties in de competentie-ijsberg en het classificeren van persoonlijkheidskenmerken in de vijf groepen persoonlijkheidskenmerken van Goldberg.

Mijn inschatting is dat er pas overeenstemming over gewenste competenties te bereiken is, als er overeenstemming over een architect-rollenstructuur is. Dat betreft

zowel de dienst die elke architect levert als de plaats in de organisatie en de taken die hij uit moet voeren. Hierbij moeten we denk ik niet streven naar een voorschrijvend rollenmodel, maar naar een inventariserend model, waarin de architectrollen vermeld worden die in organisaties gehanteerd worden.

10. Literatuur

[BMT1999]

G.J. Bergenhenegouwen, E.A.M. Mooijman, H.H. Tillema. *Strategisch opleiden en leren in organisaties*. Kluwer, 2^{de} druk, 1999.

[DV2003]

G. ten Dam, J. Vermunt. "De leerling". In N. Verloop en J. Lowyck (red.), *Onderwijskunde*. Wolters-Noordhoff, 2003. Pagina's 151-194.

[G1990]

R.L. Goldberg. "An Alternative "Description of Personality": The Big-Five Factor Structure". *Journal of Personality and Social Psychology* 59(6), 1990, pagina's 1216-1229.

[ODC2001]

J. Op de Coul. *Taken, Functies, Rollen en Competenties in de Informatica*. Ten Hagen Stam, 2001.

[VSW2004]

K. Voermans, C. Steghuis, R. Wieringa. *Architectrollen en competenties: Een enquête uitgevoerd tijdens het LAC 2004*. Intern rapport. NAF, Februari 2005.

[WV2005]

R. Wieringa, E. Vermeulen. *Competencies and Roles of ICT Architects*. Intern Rapport NAF, April 2005.

[SVW 2005a]

C. Steghuis, K. Voermans, R. Wieringa. *Competencies of the ICT Architect*. Intern Rapport NAF, Juli 2005.

[SVW2005b]

C. Steghuis, K. Voermans, R. Wieringa. *Interview Report on the Competencies of the ICT Architect*. Intern Rapport NAF, October 2005.

[RW2004]

R.J. Wieringa. "Service provision layers". GRAAL project , 2004, <http://graal.ewi.utwente.nl/whitepaper.php>.