

Een volgende stap in de ontwikkeling van architectuur

TOGAF in vogelvlucht

Danny Greefhorst

Er zijn de afgelopen tien jaar veel architectuurmethoden, technieken en raamwerken verschenen. Veel van deze methoden en technieken zijn afkomstig van adviesorganisaties en niet publiek beschikbaar. Er is echter een duidelijke trend naar open standaarden en TOGAF en ArchiMate zijn daarvan goede voorbeelden. In dit artikel wordt een overzicht gegeven van TOGAF, waarbij met name wordt ingegaan op TOGAF 9. Aanleiding is de publicatie ervan op 2 februari 2009 op de Open Group conferentie in San Diego.

Inleiding

Het enterprise architectuur vakgebied is nog relatief jong en de afgelopen 10 jaar is er veel gebeurd. Recentelijk nog is het tiende Landelijk Architectuur Congres gehouden. Het is een belangrijke constatering dat het vakgebied nog steeds in ontwikkeling is en dat we het nog lang niet met elkaar eens zijn. Met name adviesorganisaties zijn debet aan deze babylonische spraakverwarring, gemanifesteerd onder kreten als "Integrated Architecture Framework", "Dynamische Architectuur", "MArch", "Generic Enterprise Model" en "Business Informatie Planning". Waakzaamheid is geboden zodat deze onenigheid de geloofwaardigheid van architectuur niet aantast. Architectuur is immers een belangrijk hulpmiddel voor organisaties om structuur te geven aan veranderingen en architecten zijn er vooral om organisaties te helpen met hun kennis en ervaring. In de huidige praktijk wordt architectuur nog te veel tot doel op zich verheven, terwijl architectuur uiteindelijk alleen maar een middel is om de doelen van de organisatie te bereiken. Er zou meer gestreefd moeten worden naar een pragmatische en doelgerichte aanpak van architectuur.

Gegeven de bovenstaande ontwikkelingen is het belangrijk om te constateren dat er ook een aantal initiatieven op het vakgebied zijn die duidelijk bijdragen aan convergentie. Initiatieven die werken aan samenwerking, informatie-uitwisseling en professionalisering van architecten. Denk daarbij aan initiatieven zoals de Open Group, het Nederlands Architectuur Forum, Via Nova Architectura, het Genootschap voor Informatie Architecten en het Nederlands Genootschap voor Informatici. Niet onbelangrijker is dat er een aantal standaarden zijn ontstaan zoals IEEE/ANSI 1471 [IEEE, 2000], ArchiMate [Lankhorst, 2005] en TOGAF [Open Group, 2006] waardoor terminologie, notatie en methoden worden gestandaardiseerd. Dit artikel gaat met name in op TOGAF omdat dat een belangrijk convergentiepunt lijkt te worden voor architectuur. Aanleiding voor dit artikel is het verschijnen van versie 9 van TOGAF [Open Group, 2009] waarin een aantal belangrijke verbeteringen zijn doorgevoerd. Het artikel zal dan ook expliciet aandacht besteden aan de veranderingen in deze versie van TOGAF. Voor de mensen die nog niet bekend zijn met TOGAF geef ik eerst een kort overzicht van TOGAF 8.

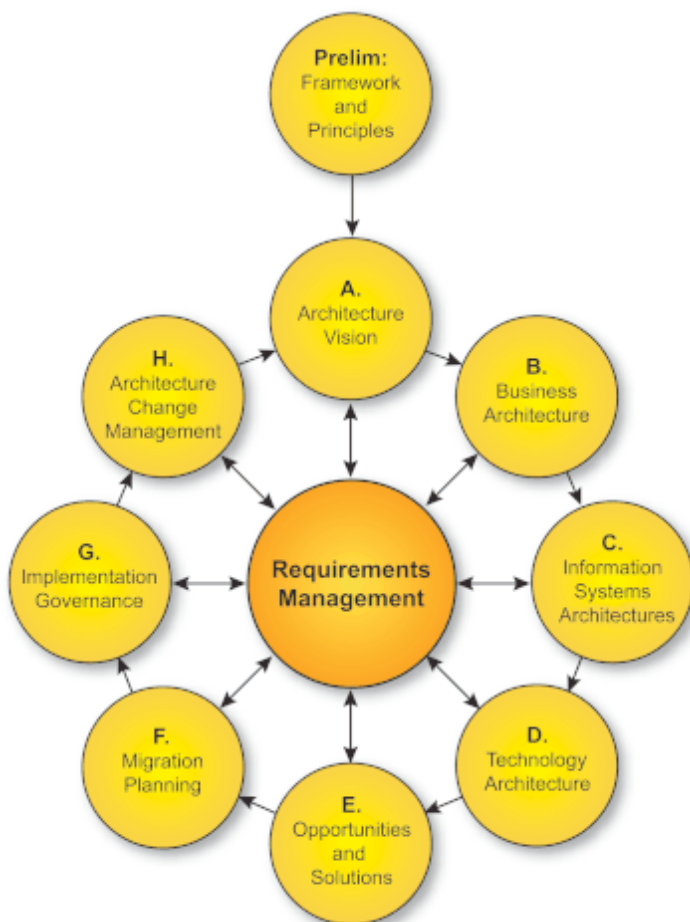
TOGAF 8

TOGAF staat voor "The Open Group Architecture Framework" en is feitelijk een verzameling van methoden, technieken en best-practices op het gebied van enterprise architectuur. In tegenstelling tot wat de naam "framework" doet vermoeden ligt de nadruk daarbij niet zozeer op de architectuurinhoud zelf, maar vooral op het organiseren en procesmatig ondersteunen van architectuur. TOGAF versie 8 bouwt voort op eerdere versies van TOGAF die teruggaan tot 1994. Nieuw in versie 8 is de uitgebreidere aandacht voor bedrijfs- en informatiesysteem architectuur. In voorgaande versies lag de nadruk nogal op de technologie architectuur. Dit is voor een belangrijk deel te verklaren door de ontstaansgeschiedenis van de Open Group. Deze heeft zicht in het verleden vooral bezig gehouden met technische standaarden zoals bijvoorbeeld de DCE standaard voor gedistribueerde systemen. Enterprise architectuur in de brede zin kan echter niet beperkt zijn tot IT alleen.

TOGAF 8 bestaat uit vier delen:

- Deel 1 (Introduction) is de inleiding en geeft een overzicht van TOGAF;
- Deel 2 (Architecture Development Method) beschrijft de architectuurmethode;
- Deel 3 (Enterprise Continuum) beschrijft een classificatie van architecturen en oplossingen;
- Deel 4 (Resource Base) bevat een verzameling van architectuur best-practices.

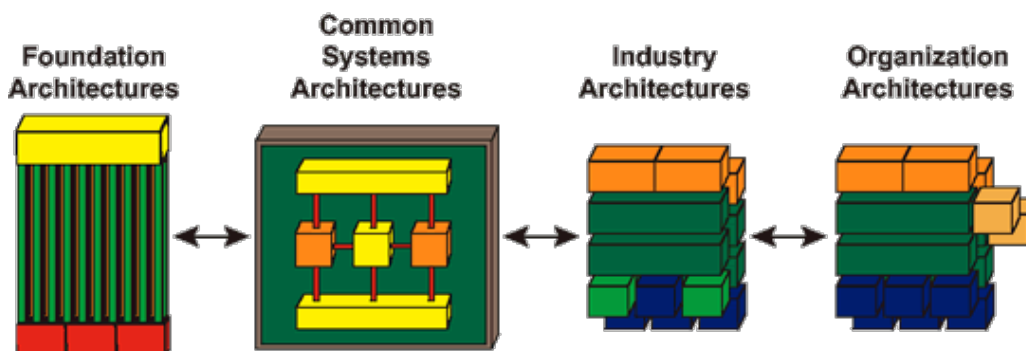
Het belangrijkste van TOGAF is eigenlijk de architectuurmethode, temeer omdat andere methoden en raamwerken hier veel minder expliciet in zijn. Deze methode biedt een stappenplan waarin is aangegeven welke activiteiten moeten worden uitgevoerd en wat hun invoer en uitvoer is. De methode geeft niet in detail aan welke inhoudelijke architectuurmodellen moeten worden gemaakt. Er worden wel op allerlei plaatsen suggesties gedaan, waarbij frequent wordt gerefereerd aan het Zachman framework [Zachman, 1987]. Ik zie de methode vooral als een handige checklist; het tot op de letter volgen van de methode leidt niet tot een pragmatische aanpak. Zoek naar wat past bij de gestelde doelen.



Afbeelding 1 Architecture Development Method

In Afbeelding 1 is de architectuurmethode grafisch weergegeven. De methode start met het bepalen van de wijze waarop de methode wordt aangepast aan de organisatie en waarin de organisatie wordt ingericht. Ook worden in deze *preliminary* fase de architectuurprincipes opgesteld die richting geven aan alle vervolgfases. In de *architecture vision* fase wordt de architectuur op hoofdlijnen bepaald, waarbij vooral gebruik gemaakt wordt van een scenario-gebaseerde werkwijze (*business scenarios*). Dit levert tevens de belangrijkste eisen die worden gesteld aan de architectuur. In de *business architecture* fase worden de bedrijfsaspecten beschreven, wat resulteert in een set van bouwblokken (*building blocks*) en views die de architectuur beschrijven. In de *information systems architecture* fase worden de *data architecture* en de *application architecture* beschreven. De volgorde is afhankelijk van de specifieke context, maar typisch worden in de data architecture op hoofdlijnen de applicaties geïdentificeerd, en worden deze in de application architecture verder uitgewerkt. Resultaat zijn wederom een verzameling bouwblokken en views. In de *technology architecture* fase wordt de vertaling gemaakt naar de technologie. Hier spelen standaarden een belangrijke rol; deze geven richting aan de uiteindelijke vertaling van architectuurbouwblokken naar oplossingsbouwblokken. In de *opportunities and solutions* fase wordt de architectuur vertaald naar een verzameling projecten en worden oplossingsbouwblokken gezocht bij de architectuurbouwblokken. In de *migration planning* fase worden de projecten geprioriteerd en gepland in tijd. In de *implementation governance* fase wordt de architectuur vertaald naar richtlijnen voor projecten, worden contracten afgesloten met de projecten en worden de projecten begeleid. In de *architecture change management* fase worden allerlei ontwikkelingen bewaakt en vertaald naar wijzigingen op de architectuur. Deze kunnen leiden tot kleine aanpassingen of tot een nieuwe cyclus van de methode. De *requirements management* fase staat centraal in alle processen en is de plaats waar alle eisen die aan de architectuur worden gesteld worden beheerd.

Het *enterprise continuum* beschrijft dat er architecturen en oplossingen op allerlei niveau's bestaan. Dit continuum bestaat op zijn beurt weer uit een *architecture continuum* (Afbeelding 2) en een *solution continuum*. De belangrijkste boodschap daarbij is dat er niet alleen organisatie-specifieke architecturen zijn (*organisation architecture*), maar ook meer generieke en herbruikbare architecturen. Deze kunnen specifiek zijn voor een bepaalde sector (*industry architecture*), maar ook sector-onafhankelijk (*common systems architecture*). In het meest extreme geval is het een generieke classificatie van elementen en spreekt TOGAF over een *foundation architecture*. Voor elk van deze vormen van architectuur zijn er ook oplossingen die deze architecturen ondersteunen, bijvoorbeeld een standaard softwarepakket voor een specifieke sector. Andersom geredeneerd geven de architecturen richting aan de oplossingen. Het enterprise continuum biedt feitelijk een classificatie van architecturen en oplossingen die je zou kunnen gebruiken voor het organiseren van een architectuurrepository. TOGAF spreekt over een *virtual repository*.



Afbeelding 2 Architecture Continuum

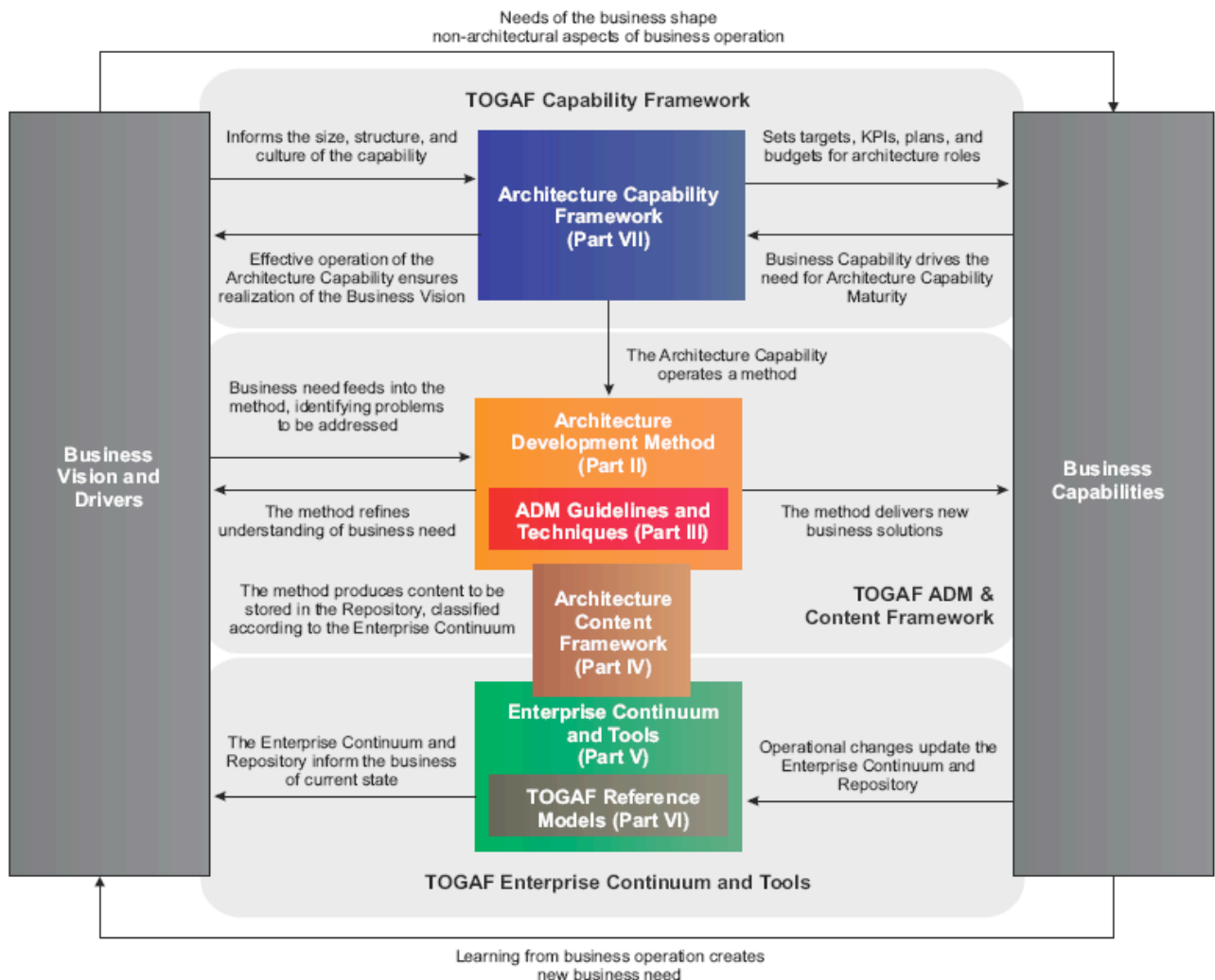
De *resource base* beschrijft een aantal best-practices op het gebied van architectuur. Er zijn twee belangrijke groepen van hoofdstukken in deze resource base. De eerste groep heeft met name betrekking op de inrichting van architectuur in de organisatie. Hier lees je over wat architectuur governance is, hoe je een architectuur board inricht, welke competenties architecten zouden moeten hebben en hoe om te gaan met architectuur compliance en architectuurcontracten. De andere groep gaat meer in op architectuurmodellen zelf. Zo geeft

het een goed beeld van wat architectuurprincipes zijn en geeft het hiervoor een goede startset. Daarnaast gaat het in op wat architectuurviews zijn en geeft het een beeld van de issues die in deze views geadresseerd zouden moeten worden. Zoals eerder aangegeven blijft het daarbij in het midden wat er dan precies in een view staat. Ook lijkt de tekst op een aantal plaatsen niet echt bijgewerkt op nieuwe inzichten.

TOGAF 9

Op 2 februari 2009 is TOGAF 9 aangekondigd op de Open Group conferentie in San Diego. Deze nieuwe versie van TOGAF blijkt een aantal belangrijke verbeteringen te zijn ondergaan ten opzichte van de vorige versie. We zullen in dit artikel ingaan op de veranderingen. Ervan uitgaande dat dit je nieuwsgierigheid heeft opgewekt verwijst ik graag naar de web-site van de Open Group voor de details (<http://www.opengroup.org/>). Dit is een belangrijk voordeel van TOGAF; het is gratis beschikbaar en als organisatie kun je ook meehelpen aan het verder verbeteren van TOGAF.

De eerste belangrijke verbetering is in de structuur van TOGAF (zie Afbeelding 3). Er zijn een aantal nieuwe delen gedefinieerd die een duidelijker herkenbare inhoud hebben en die ook los van de andere delen kunnen evolueren. Zo zijn de tips en technieken voor het inrichten van de methode in een separaat deel terecht gekomen, alsook best-practices rondom het inrichten van architectuur in een organisatie. Ook de referentie architecturen hebben een eigen deel gekregen. Een geheel nieuw deel is het content framework waar we in de volgende paragraaf dieper op in zullen gaan. Verder zijn ook de relaties tussen de verschillende delen inzichtelijker gemaakt wat bijdraagt aan een beter begrip van de architectuur van TOGAF.

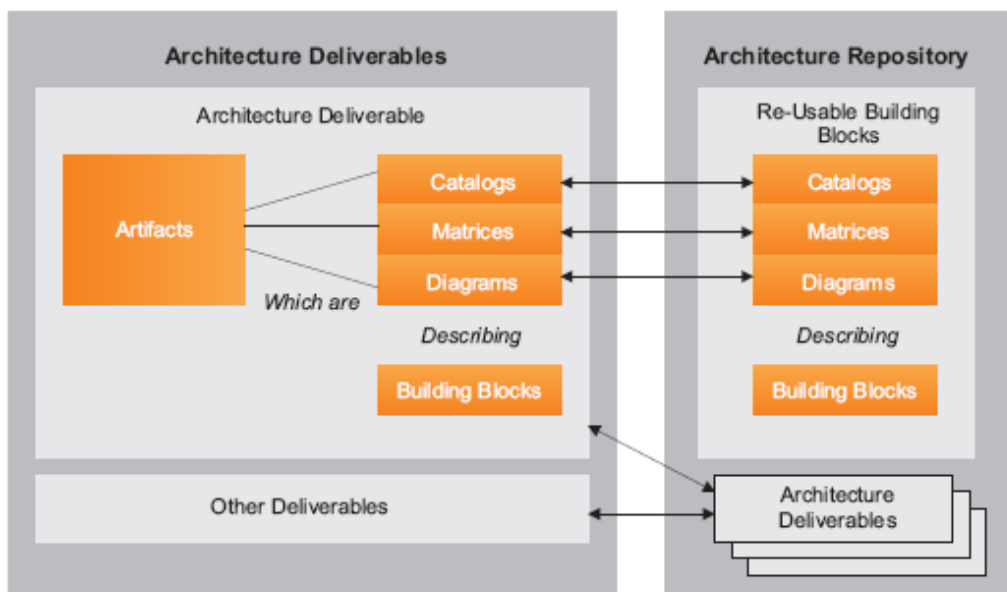


Afbeelding 3 Inhoudsopgave van TOGAF 9

Verreweg de belangrijkste toevoeging is het *content framework* dat definieert waar een architectuur feitelijk uit bestaat. Zoals eerder in dit artikel aangegeven was dit een erg zwak punt in de vorige versie. Eigenlijk werd niet goed duidelijk welke specifieke modellen en views je moet maken op een specifiek moment en wat daarvan de precieze inhoud is. Ook was het onvoldoende helder wat de precieze invoer en uitvoer van fasen en stappen in de methode was. Oorzaak daarvan is ondermeer een inconsistent taalgebruik, onduidelijke definities en onvolledigheden.

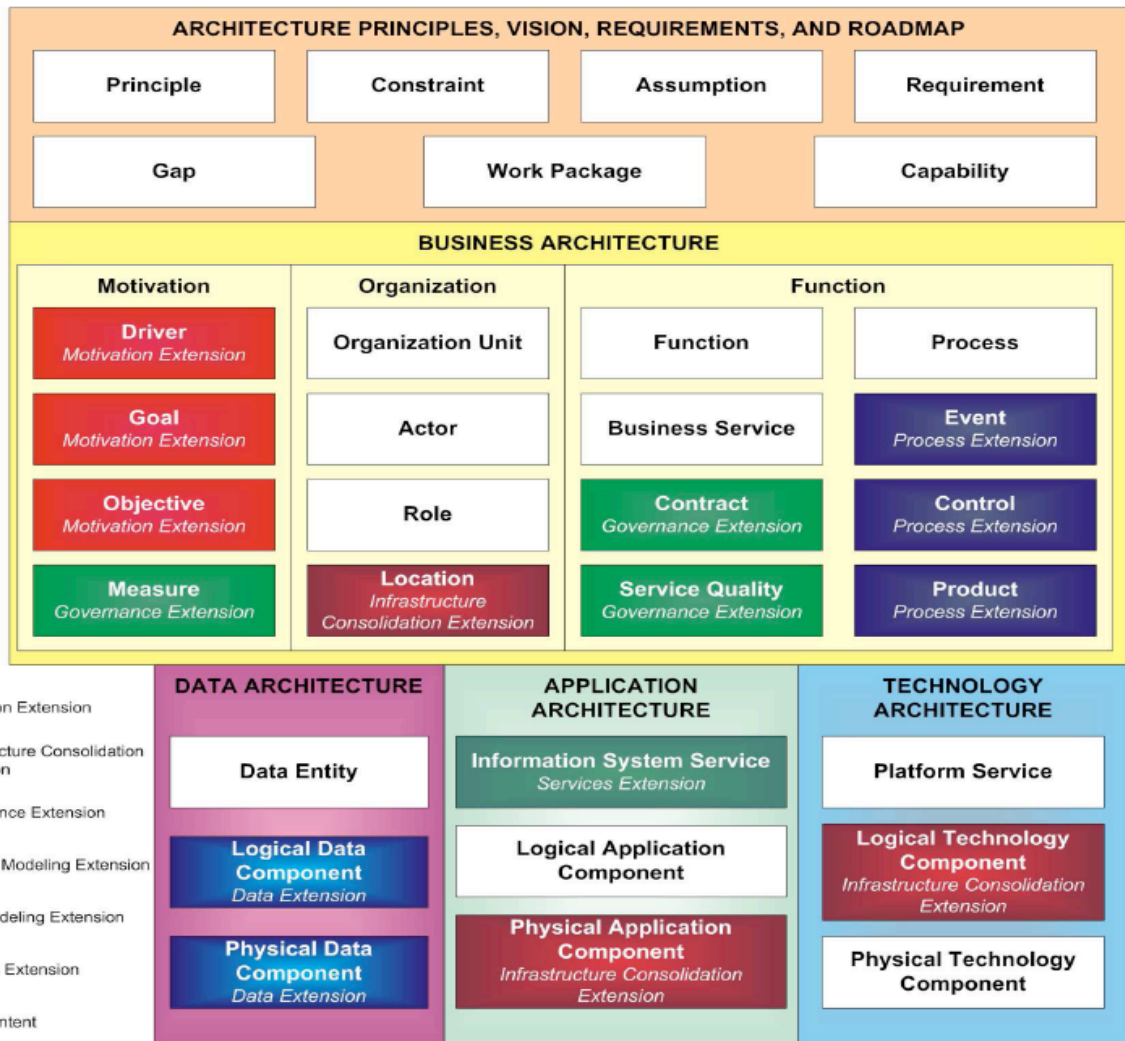
Het content framework

Laten we eens kijken wat de belangrijkste onderdelen zijn van het *content framework*. Afbeelding 4 geeft een overzicht van deze onderdelen en hun relatie. De eenheden die formeel worden opgeleverd in een architectuurtraject zijn *deliverables*. Denk daarbij aan documenten waar de opdrachtgever ook zijn handtekening op zet. Deze deliverables bestaan op hun beurt weer uit *artefacten*. Dit zijn individuele modellen die zich kunnen manifesteren als een diagram, maar ook als een lijst (*catalog*) of matrix. In deze structuur zien we al snel de oorsprong van het content framework terugkomen; het Integrated Architecture Framework (IAF) van Capgemini. In deze aanpak wordt namelijk veel gewerkt met matrices die de relaties tussen allerlei architectuurelementen aangeven en daarmee een stuk traceerbaarheid bewerkstelligen. Artefacten beschrijven bouwblokken; de eenheden waarin architectuurelementen worden gedefinieerd. Dit was in de vorige versie een nogal abstract begrip. In versie 9 wordt het duidelijker; een bouwblok is een concept in het architectuur metamodel (zie volgende paragraaf) en kan daarmee feitelijk heel veel verschillende dingen zijn. Het zijn de eenheden die je registreert in je architectuurrepository; een begrip dat ook in deze versie is toegevoegd. Daarmee is het wel weer zo'n algemeen begrip dat je je afvraagt waarom er zoveel ophef over wordt gemaakt; er is immers wel weer een heel hoofdstuk over.



Afbeelding 4 Bouwblokken, artefacten en deliverables

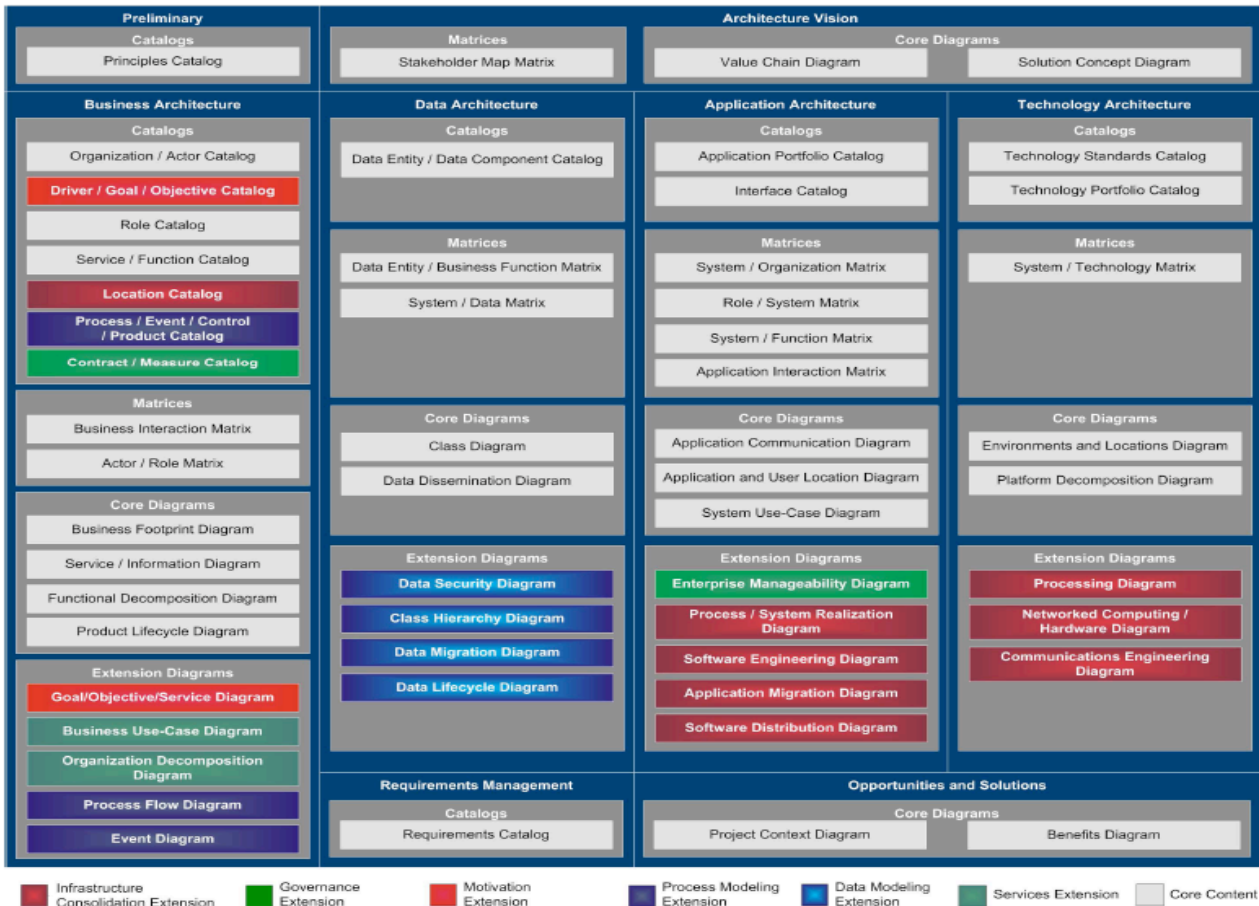
De basis voor het content framework is het *content metamodel* waarin alle relevante concepten zijn gedefinieerd, inclusief hun relatie (zie Afbeelding 5). Voor alle architectuurdomeinen zijn deze concepten uitgewerkt en wordt zelfs een voorzet gegeven voor de relevante attributen die je in een architectuurrepository zou willen administreren. Ook in het metamodel herkennen we heel duidelijk het IAF raamwerk van Capgemini. Het is lovenswaardig dat Capgemini zoveel van zijn eigen intellectual capital in TOGAF heeft verwerkt. Ik vind het wel jammer dat hiermee tegelijk een concurrerend begrippenkader voor ArchiMate in TOGAF is opgenomen. Deze enterprise architectuurtaal was juist afgelopen jaar geadopteerd door de Open Group. De hoop is dat een volgende versie van TOGAF alle verschillen tussen ArchiMate en het TOGAF metamodel zijn gladgestreken. Gelukkig zit er in ieder geval geen notatie in TOGAF, zodat daar in ieder geval niet in gedivergeerd wordt.



Afbeelding 5 Content metamodel (bouwblokken)

Wat verder opvalt in het metamodel is dat er gekozen is voor een gelaagde structuur. Zo is er een basismetamodel waar alle kernbegrippen in voorkomen en zijn er specifieke uitbreidingen voor gebieden zoals motivatie, infrastructuur, governance, procesmodellering, data modellering en service oriëntatie. Alhoewel het doel van een aantal concepten daarmee wel helderder wordt vind ik het ook wat kunstmatig om deze opdeling aan te brengen. Uiteindelijk is het één geïntegreerd metamodel. Ook vind ik het bijvoorbeeld merkwaardig dat een begrip als *business service* feitelijk ook de semantiek van een *application system service* heeft als je de service oriëntatie extensie niet gebruikt. Alsof een concept van betekenis kan veranderen. Om nog maar niet te spreken over een vreemde inconsistentie van naamgeving; waarom levert een *logical application component* niet gewoon een *application service*?

Een ander belangrijk deel van het content framework is de definitie van artefacten. Daarbij wordt verwezen naar het IEEE model en worden artefacten gelijk gesteld aan views. Eerder gaven we al aan dat er drie soorten artefacten zijn: diagrammen, matrices en lijsten. Dit zien we ook terug in de lijst van voorgedefinieerde artefacten (zie Afbeelding 6). Deze lijst (of framework) is wederom een sterke verbetering van de lijst van views zoals gedefinieerd in TOGAF 8. Die waren niet helder gedefinieerd. Voor alle artefacten in TOGAF 9 is een alinea met tekst waarin helder wordt wat het doel is van het artefact en waar het uit bestaat. In de verschillende fasen en stappen van de architectuurmethode wordt ook expliciet gerefereerd aan deze artefacten, waardoor de methode een veel betere basis krijgt. Jammer is dat deze artefacten afwijken van de viewpoints in ArchiMate. Ook is het vreemd dat in de tekst nog steeds wordt ingegaan op een aantal viewpoints in TOGAF 8 die geen plaats hebben in Afbeelding 6 zoals bijvoorbeeld de *system engineering view*.

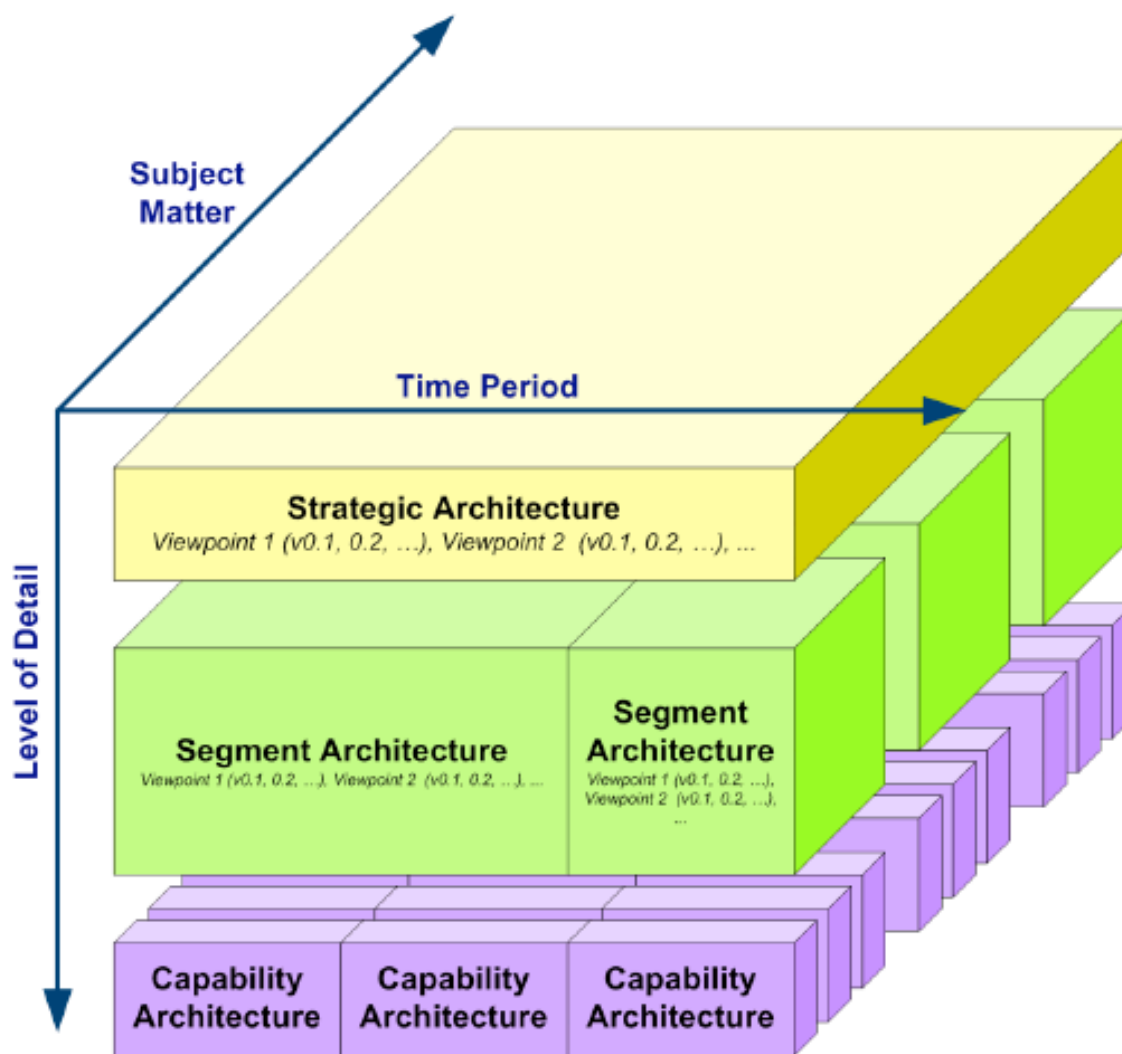


Afbeelding 6 Artefacten

De deliverables worden veel beter gedefinieerd in TOGAF 9. Een aantal begrippen die duidelijk ontbraken zijn daarin toegevoegd zoals bijvoorbeeld *architecture definition document*, *architecture repository* en *architecture requirements specification*. De deliverables zijn de invoer en uitvoer van de fasen en stappen in de methode. Door een betere definitie van de deliverables wordt de methode ook begrijpelijker. Zo waren bijvoorbeeld de requirements in TOGAF 8 nogal impliciet aanwezig en onduidelijk gepositioneerd.

Partitionering van architecturen

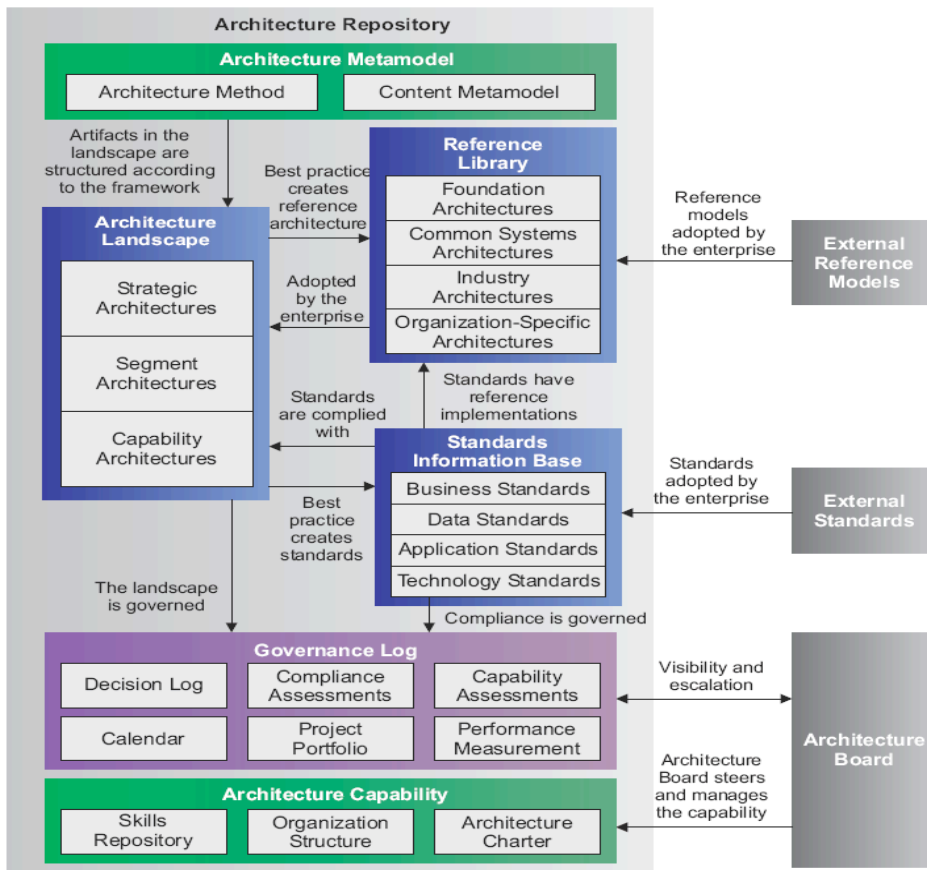
Een belangrijke toevoeging in TOGAF 9 is een verder onderscheid in soorten architecturen. Er is onderkend dat er in een typische organisatie allerlei architecturen bestaan en dat je vantevoren goed moet nadenken over welke architecturen je wilt onderscheiden en welke relaties deze met elkaar hebben. TOGAF noemt dat het *partitioneren* van architectuur en daar is een heel hoofdstuk aan gewijd. Daarin worden ondermeer de belangrijkste dimensies geïdentificeerd op basis waarvan je zou willen partitioneren: het onderwerp, het viewpoint, het detailniveau, het abstractieniveau en de accuraatheid. Ook wordt aangegeven hoe deze dimensies kunnen worden gehanteerd bij het onderverdelen van het architectuurlandschap (*architecture landscape*). Dit laatste is overigens ook een nieuw begrip in TOGAF 9 wat zoveel betekent als de architectuurniveau representatie van de organisatie. De typische vormen van architectuur die je kunt onderkennen zijn weergegeven in Afbeelding 7. Hierin wordt een onderscheid gemaakt tussen *strategic architectures*, *segment architectures* en *capability architectures*, die zich vooral onderscheiden qua detailniveau. Strategic architectures beschrijven op hoog niveau de gehele organisatie. Segment architectures gaan in op specifieke inhoudelijke deelgebieden. Capability architectures zijn architecturen voor een specifieke oplossing. Het is onduidelijk waarom niet meer gangbare termen als "enterprise architecture", "domain architecture" en "solution architecture" worden gehanteerd.



Abbeelding / Partitionering van het architectuurlandschap

Partitionering heeft niet alleen betrekking op het architectuurlandschap, maar ook op referentie architecturen. Dit is feitelijk het architecture continuum zoals we het al kennen vanuit TOGAF 8, alleen wordt nu aangegeven dat dit *reference models* zijn (er wordt in de tekst ook wel over *reference architectures* gesproken). Interessant is dat gesteld wordt dat ook *organisation-specific architectures* in die categorie vallen. Daarmee is er in TOGAF een veel duidelijker onderscheid tussen architecturen met hele specifieke keuzen (strategisch, segment en capability) en herbruikbare referentie architecturen (foundation, common systems, industry, organisation-specific) ontstaan. Dit onderscheid zie je ook terug in het hoofdstuk dat in gaat op wat een architectuurrepository is. Deze repository bevat beide categorieën van architecturen (zie Afbeelding 8).

Overigens zit er in deze repository eigenlijk alle voor architectuur relevante informatie zoals het metamodel, de standaarden (*standards information base*), governance gerelateerde informatie zoals besluiten, alsook informatie over de inrichting van architectuur in de organisatie. Er wordt in TOGAF 9 dus niet langer over het enterprise continuum gesproken als een *virtual repository*; in plaats daarvan hebben we een echte architectuurrepository. Deze repository staat centraal in alle fasen en activiteiten; resultaten worden er in opgeslagen en in vervolgfases hergebruikt.



Abbeelding 8 Architectuurrepository

Vernieuwingen in de architectuurmethode

De architectuurmethode zelf is vrij stabiel gebleven. Op het hoogste niveau is deze feitelijk ongewijzigd. De aanpassingen zitten wat dieper onder deze structuur. De belangrijkste verbetering is de onderbouwing van de fasen en stappen met zaken uit het content framework. Een andere belangrijke verbetering is een gelijk stappenplan voor alle architectuur domeinfasen (business, data, application en technology). De stappen in al deze fasen zijn nu:

1. Selecteren van referentiemodellen, viewpoints en tools
2. Het ontwikkelen van de baseline architectuur
3. Het ontwikkelen van de target architectuur
4. Het uitvoeren van een gap analyse (verschil tussen baseline en target)
5. Het definiëren van een roadmap van componenten
6. De impact op andere delen van het architectuurlandschap bepalen
7. Een review door relevante betrokkenen uitvoeren
8. Het afronden van de architectuur
9. Het opstellen van het architectuurdefinitie document

Daarnaast is de tekst zelf verder verrijkt en wordt je dus beter geholpen in het uitvoeren van de architectuurmethode. Uitbreidingen zitten met name in de preliminary fase en de opportunities & solutions fase.

Richtlijnen en technieken voor de architectuurmethode

Alhoewel de methode zelf dus niet echt is veranderd is er wel een heel nieuwe deel bijgekomen waarin richtlijnen en technieken worden gegeven die kunnen worden toegepast in de architectuurmethode. Hier staan een aantal bekende hoofdstukken in zoals die met architectuurprincipes en business scenarios, maar er zijn ook een aantal geheel nieuwe hoofdstukken.

Wat opvalt is dat het takenpakket van de architect breder lijkt te worden. We zien een aantal technieken die voor project managers al erg bekend zijn zoals *stakeholder management* en *risico management*. Het is ook als architect natuurlijk belangrijk om te weten wat hoe je met betrokkenen en met risico's moet om gaan. Ook zien we technieken die in het verleden meer het werkgebied van business consultants was. Zo is er een soort van organisatie assessment aanpak opgenomen in TOGAF: de *business transformation readiness assessment*. Het idee ervan is dat je voordat je overgaat tot het implementeren van veranderingen een goed beeld hebt in hoeverre een organisatie klaar is voor de veranderingen. De aanpak gaat uit van factoren die de veranderbereidheid (*readiness*) van een organisatie bepalen. Deze factoren moeten worden bepaald en uitgewerkt, waarna de feitelijke scoring wordt uitgevoerd. Op basis van de scoring bepaal je de noodzakelijke acties en vertaal je deze naar een migratieplan. Ook de techniek van *capability based planning* is een uitgebreidere, meer strategische blik op de rol van de architect. Het is een andere manier om te kijken naar veranderingen die nodig zijn in een organisatie. Het idee is om zogenaamde *capabilities* te definiëren die door de hele organisatie heen snijden, maar fundamenteel zijn om de doelen van de organisatie te bereiken. Denk daarbij bijvoorbeeld aan Sarbanes-Oxley Compliance. De resulterende capabilities zijn een alternatieve bron van input voor de architectuurmethode. Het is een interessant idee om meer vanuit capabilities naar een organisatie te kijken. Het wordt het alleen niet echt duidelijk hoe je tot de juiste capabilities komt.

Er worden een aantal technieken besproken die erg dicht op de architectuurmethode zelf zitten. Zo wordt de architect allerlei technieken aangereikt die helpen bij het uitvoeren van migratie planning. In het bijzonder vind je allerlei soorten matrices die je kunt definiëren en die je inzicht geven in de stappen die moeten worden gezet, wanneer ze moeten worden gezet en hoe de architectuur evolueert in tijd. Projecten moet je afzetten tegen de waarde die ze leveren en de risico's die ermee geassocieerd zijn. Op basis daarvan kan worden bepaald of projecten moeten worden doorgezet. De architect wordt ook geholpen bij het bepalen van de precieze iteraties in de architectuurmethode, wat één van de belangrijkste kenmerken ervan is. Globaal wordt een onderscheid gemaakt tussen *baseline first* en *target first*. Dit geeft aan of je met de huidige of gewenste situatie start en wat daarvan de impact is op de verschillende fasen in de architectuurmethode. Een iteratieve aanpak voor architectuur is mijn inziens essentieel om deze pragmatisch te houden; het zorgt ervoor dat je snel met resultaten kunt komen en snel inzicht krijgt in de gebieden die verdere verdieping vragen.

TOGAF 9 geeft nu inzicht in de soorten architectuurprojecten (*architecture engagements*) die je kunt uitvoeren. Daarbij worden drie fundamentele klassen onderkend: het identificeren van veranderingen, het definiëren van veranderingen en het implementeren van veranderingen. Hierin wordt duidelijk dat architectuur zich kan bewegen van strategisch tot operationeel niveau. Dit heeft een duidelijke relatie met de soort architecturen (strategic, segment, capability) en de iteratieaanpak die wordt gekozen.

Specifieke gebieden die voor architecten erg belangrijk zijn worden verder uitgewerkt. Een belangrijk gebied voor architecten is security. In een separaat hoofdstuk wordt ingegaan op security architectuur en de relatie die dat heeft met de verschillende fasen in de architectuurmethode. Zo worden per fase beschreven welke activiteiten relevant zijn en welke security gerelateerde invoer en uitvoer relevant is.

Een ander belangrijk gebied voor architecten is applicatie-integratie. De basis hiervoor zijn de eisen die eraan gesteld worden; *interoperability requirements*. We leren dat er verschillende soorten interoperability requirements bestaan; bijvoorbeeld op verschillende lagen van een applicatie. Daarnaast wordt een techniek aangereikt waarbij de uitwisseling tussen partijen wordt geclassificeerd op basis van de mate van integratie. De resulterende matrix lijkt wel wat kunstmatig.

De service-georiënteerde invalshoek is een separaat hoofdstuk beschreven. Dit gaat ondermeer in op het onderscheid tussen *business-led SOA* en *developer-led SOA*. Er wordt aangegeven wat er in de verschillende fasen aan SOA gerelateerde aandachtspunten zouden moeten worden opgepakt. Tenslotte wordt vrij uitgebreid stilgestaan bij het concept *service-contract*; een manier om afspraken tussen aanbieders en afnemers van services te formaliseren.

Overige veranderingen

Meer kleinschalige aanpassingen aan TOGAF zijn de toevoeging van een *document categorization model* waarin wordt aangegeven hoe de verschillende delen van TOGAF los van elkaar evolueren en wat de status ervan is. Ook is er nu een uitgebreidere lijst van definities van begrippen.

Er zijn ook zaken verwijderd uit TOGAF zoals de beschrijving hoe om te gaan met het ontwikkelen van views, de standards information base, de case studies en de relatie met andere raamwerken. Verder zijn er ook veel hoofdstukken (vrijwel) ongewijzigd zoals de hoofdstukken over requirements management, architectuurprincipes, architectuurpatronen, business scenarios, tools, het technical reference model, het Integrated Information Infrastructure Reference Model en veel van de hoofdstukken over de organisatie van architectuur, die nu in het *architecture capability framework* terecht is gekomen.

De certificatie van TOGAF 9 verdient ook aandacht. Architecten die TOGAF 8 gecertificeerd zijn, zijn niet automatisch ook TOGAF 9 gecertificeerd. Zij moeten opnieuw examen doen. Zij hebben wel het voordeel dat dit open boek mag. Voor de nieuwkomers is er nu een onderscheid tussen level 1 en level 2 certificering, waarbij voor de tweede ook een zwaarder examen geldt.

Conclusies

Ik heb in dit artikel een overzicht gegeven van TOGAF, waarbij ik met name ben ingegaan op de nieuwe inhoud van TOGAF 9. Het is duidelijk dat TOGAF 9 een meer volwassen aanpak biedt voor architectuur. Er zijn veel nieuwe zaken toegevoegd en de methode is een stuk beter onderbouwd, waardoor het een veel beter fundament heeft. Dit alles maakt TOGAF 9 tot een serieus raamwerk waarin iedere professionele architect eigenlijk zou moeten zijn gecertificeerd.

Natuurlijk zijn er ook nog zaken op te merken aan TOGAF 9; als architect ben je nu eenmaal kritisch. De belangrijkste omissie is dat het content framework nog niet in lijn is met ArchiMate. Ik kan dan ook niet wachten tot ArchiMate geïntegreerd is in TOGAF. Daarnaast vind ik dat er toch ook nog veel oude inhoud te vinden is die eigenlijk niet meer in lijn is met de nieuwe delen van TOGAF. Ik hoop dan ook dat deze delen er in een volgende versie gewoon uit zijn. Tenslotte zou ik graag meer ondersteuning zien in het maken van specifieke architectuurinhoud.

Danny Greefhorst

ArchiXL

dgreefhorst@archixl.nl

Referenties

- | | |
|--------------------|--|
| [IEEE, 2000] | IEEE Std 1471-2000: "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems", 2000. |
| [Open Group, 2006] | Open Group: "The Open Group Architecture Framework", version 8.1.1., ISBN: 1-931624-62-3, 2006. |
| [Open Group, 2009] | Open Group: "The Open Group Architecture Framework", version 9, ISBN: 9789087532307, 2009. |
| [Lankhorst, 2005] | M.M. Lankhorst et al.: "Enterprise Architecture at Work", Springer, 2005. ISBN 3-540-24371-2. |
| [Zachman, 1987] | J.A. Zachman: "A framework for information systems architecture", IBM Systems Journal 1987. |