

Tools voor architectuur

Ria van Rijn

In deze white paper besteden we aandacht aan tools, die het maken en beheren van architectuurproducten kunnen ondersteunen. Allereerst wordt er aandacht besteed aan de voor- en nadelen van het gebruik van tools. Daarna komen een aantal eisen, die aan architectuur tools moeten worden gesteld, aan de orde.

Om de eisen aan een architectuurtool beter te onderbouwen besteden we eerst aandacht aan de standaard van ISO/IEC/IEEE, die de onderdelen waaruit een architectuur moet bestaan, definieert. Deze onderdelen moeten natuurlijk ook allemaal in een tool terug te vinden zijn. Een ander belangrijk vereiste voor een tool is natuurlijk de modelleertaal, die het ondersteunt. Ook andere eisen, die aan tools gesteld moeten worden, worden verder uitgewerkt.

Tenslotte wordt aandacht besteed aan twee instrumenten, waarmee een eventuele toolselectie kan worden voorbereid: de Gartner Magic Quadrant voor enterprise architectuur tools en de Enterprise Architecture Tool Selection Guide.

1. Voordelen van een architectuur tool

Architecturen hebben betrekking op verschillende domeinen. Meestal worden de domeinen business, organisatie, informatie en technologie onderscheiden. En ook verschillende invalshoeken: bestemmingsplan, domeinarchitectuur en project start architectuur. Naarmate je meer domeinen hebt beschreven in je architectuur, of meer invalshoeken gebruikt en van meer verschillende gezichtspunten gebruikmaakt om de architectuur aan specifieke belanghebbenden te presenteren, zal de behoefte aan een ondersteunende architectuur tool toenemen.

Het voordeel van het gebruiken van een architectuur tool ligt in de eerste plaats in de mogelijkheid de gemodelleerde concepten uit de verschillende domeinen met elkaar te verbinden. Ieder domein kent vanuit het verleden zijn eigen concepten en notatiewijzen en vaak ook zijn eigen gespecialiseerde tools. Het doel van architectuur is nu juist een consistent en coherent geheel van principes en modellen op te stellen, dat aangeeft hoe business, organisatie, informatievoorziening en technologie ontworpen en gerealiseerd moeten worden. Het is lastig als je de coherentie en consistentie moet bewaken tussen modellen als je werkt met een apart tool voor ieder domein, of - erger nog - in een veelheid van Visio en PowerPoint platen, die opgenomen zijn in verschillende vaak omvangrijke documenten.

Snel genereren van gezichtspunten

Het gebruik van één tool voor het aanbrengen van samenhang over alle domeinen heeft dan grote voordelen. Het voorkomt, dat het inzicht in de samenhang tussen de verschillende domeinen slechts in de hoofden van enkele ervaren architecten zit. Deze kennis is dan erg kwetsbaar. Het kost bovendien steeds veel tijd om die samenhang over te dragen aan anderen. Een ander voordeel van het gebruik van een tool is, dat overzichten of specifieke gezichtspunten snel gegenereerd kunnen worden. Het belang daarvan is weer, dat de architectuurfunctie snel antwoord kan geven op de vragen van de diverse belanghebbenden, bijvoorbeeld het management of van technisch beheer. Bij het vaststellen van zaken als de scope of de prioriteit van een project kan dit van groot belang zijn.

Ook het visualiseren van bepaalde onderwerpen is eenvoudiger door het gebruik van een architectuur tool. Als de platen met de gezichtspunten voor de verschillende groepen belanghebbenden steeds met de hand moeten worden gemaakt, kost dat relatief veel tijd. Het

gevolg daarvan is, dat het aantal visualisaties vaak beperkt blijft tot een beeld voor de beslissers en wat beelden voor de ontwerpers. Andere gezichtspunten worden zelden gemaakt. Dat is jammer, want juist het snel kunnen bieden van duidelijke gezichtspunten aan specifieke belanghebbenden kan sterk bijdragen aan het succes van de architectuurfunctie en het draagvlak voor werken onder architectuur. Als een architect direct vanuit zijn of haar tool bijvoorbeeld aan een proceseigenaar kan laten zien welke andere processen zijn gegevens gebruiken, zal dat direct gevolgen hebben voor de toegevoegde waarde van en de waardering voor architectuur.

Standaardiseren

Het gebruik van één tool over de verschillende architectuurdomeinen heen heeft bovendien tot effect dat er binnen de architectuurfunctie zelf meer gestandaardiseerd moet worden. Binnen de verschillende architectuur domeinen worden vaak verschillende concepten, semantiek en notatiewijzen gehanteerd. Ze liggen natuurlijk dicht bij elkaar, maar zijn toch vaak nét zo anders, dat het moeilijk is het er met alle architecten eens te worden over een gezamenlijke standaard. Het gebruik van een enterprise architectuur tool dwingt de architecten afspraken te maken over het gebruik van de concepten, semantiek en notatiewijze binnen het tool. En dit is natuurlijk ook weer bevorderlijk voor het bewaken van consistentie en coherentie van de enterprise architectuur.

Een ander voordeel is, dat een individuele domeinarchitect niet zelf verantwoordelijk is voor het aanbrengen van samenhang met de andere domein architecturen. Dit betekent namelijk veel handmatig werk. Eventueel gebruikte tools worden immers vaak alleen binnen het eigen domein gebruikt. Het vereist ook veel communicatie met de architecten om de wijzigingen in de andere domeinen bij te houden en dat kost dan weer veel tijd.

Onderhoud en versiebeheer

Dit leidt direct tot een volgend voordeel van tools: het onderhouden en (versie)beheren van architecturen wordt veel eenvoudiger. Neem als voorbeeld het wijzigen van namen van concepten, wat zeker in de fase, dat je aan het ontwerpen bent, erg vaak voorkomt. In een tool waarin de gebruikte concepten in een database worden opgeslagen (een repository) hoeft de naam meer één keer gewijzigd te worden en zal de nieuwe naam automatisch gebruikt worden in ieder model, waarin het concept gebruikt is. Werk je met verschillende tools of met Visio of PowerPoint, dan kost het erg veel tijd om alle teksten en modellen te wijzigen en dan nog liggen fouten voor de hand.

Het gebruik van Visio en PowerPoint voor het modelleren heeft nog een groot nadeel: het is vaak niet goed mogelijk eerder gemaakte modellen aan te passen of uit te breiden naar een ander gezichtspunt. Vaak is het even snel om maar weer een hele nieuwe plaat te maken. Eventuele nieuwe inzichten worden weergegeven in deze nieuwe plaat, maar niet doorgevoerd in de achterliggende, oudere platen. Die moeten bovendien vaak als zodanig in stand blijven, omdat ze gebruikt zijn in eerdere, formeel goedgekeurde documenten.

Analyseren

Niet alleen onderhoud en beheer zijn relatief eenvoudiger met een tool voor architectuur. Een tool maakt het ook mogelijk een vastgelegde architectuur beter te analyseren. Een groot voordelen van het gebruik van een architectuur tool is, dat het controleren van coherentie en consistentie gedeeltelijk geautomatiseerd kan plaatsvinden. Dit is alleen mogelijk, als alle vastgelegde concepten en relaties opgeslagen worden in één repository.

Voorbeelden van analyses op coherentie en consistentie zijn:

- het afleiden van indirecte relaties en indirecte afhankelijkheden tussen verschillende concepten;
- het signaleren van concepten, die geen relaties hebben, of die niet gebruikt worden in modellen;

- het genereren van allerlei matrices, die inzicht in wederzijdse relaties geven, bijvoorbeeld van het gebruik van applicaties door processtappen.

Voorbeelden van inhoudelijke analyses zijn een impact analyse of een gap analyse. Bij een impact analyse wordt gekeken, welke concepten er geraakt worden door een bepaalde verandering. Bijvoorbeeld wanneer een applicatie vervangen moet worden, kan snel geanalyseerd worden welke processen, diensten, interne en externe actoren, data objecten, servers, verbindingen, etc. door een dergelijke vervanging geraakt worden. Een dergelijke analyse kan de besluitvorming over de vervangingsvraag goed ondersteunen. Bovendien kan bijvoorbeeld de scope van het project, dat de vervanging moet gaan realiseren, er goed mee vastgesteld worden.

Een ander voorbeeld is een gap analyse. Dat is een analyse van de voorzieningen in termen van architectuur concepten, die gerealiseerd moeten worden om van de ene gedefinieerde toestand naar de volgende gedefinieerde toestand te komen.

Principes

Bepaalde tools maken het ook mogelijk principes vast te leggen in de tool en bepaalde concepten vervolgens weer te relateren aan één of meer principes. Hiermee kan in de platen zichtbaar gemaakt worden welke architectuur concepten voldoen aan welk principe, of welke concepten welk principe realiseren. Maar ook kan zichtbaar gemaakt worden welke principes (nog?) geen enkele relatie met architectuur concepten heeft.

Visualisaties

Visualisatie kunnen met de meeste tools gegenereerd worden. Een goede tool biedt de mogelijkheid om niet alleen de inhoud (welke onderdelen van de architectuur getoond worden) maar ook de presentatie (hoe de onderdelen getoond worden) aan te geven en vervolgens de bijbehorende plaat te genereren. Alle, onderling soms sterk verschillende, visualisaties zijn zo gebaseerd op één architectuurmodel. Daardoor kunnen belanghebbenden in hun eigen taal worden aangesproken over een onderwerp, dat hen bezighoudt, zonder heel veel extra inspanning voor de architecten. De platen zijn snel te genereren en de verschillende platen zijn onderling consistent.

Dit leidt natuurlijk tot een grote verbetering in de communiceerbaarheid van de architectuur. Door de antwoorden op concrete vragen van bijvoorbeeld projectleiders, technisch beheer of management snel te kunnen presenteren in een voor hen begrijpelijke vorm, gaat architectuur leven in de organisatie. De behoefte aan architectuur zal hierdoor ook toenemen.

Al met al leidt het gebruik van een tool ertoe, dat een architect veel minder tijd kwijt is met zaken als visualisaties, het beheren en onderhouden van de architectuur en met het bewaken van coherentie en consistentie, omdat deze activiteiten gedeeltelijk geautomatiseerd zijn. Daardoor kan meer tijd gestoken worden in het daadwerkelijk ontwerpen van goede architecturen en het afstemmen - met geschikte visualisaties - met de verschillende belanghebbenden.

Een waarschuwing is wel op zijn plaats. Het gebruik van architectuurtools is pas nuttig als de architectuur een zekere mate van volwassenheid heeft. De reden hiervoor is, dat er eerst een zeker volume aan architecturen moet zijn, wil het gebruik van tools toegevoegde waarde hebben. Als de architectuurfunctie toegevoegde waarde biedt, en je dus veel architecturen produceert, raak je vanzelf op het punt, waarop je extra hulpmiddelen nodig hebt, om tijdig aan de vraag vanuit de organisatie te voldoen.

Begin je te vroeg met een tool, dan heb je meestal nog te weinig architectuur om in de tool op te nemen. Daardoor behaal je ook niet de hier genoemde voordelen van het gebruik van een tool. Dat neemt niet weg, dat het toch handig kan zijn om je eigen werk eenvoudiger mee te maken.

2. Inhoudelijke eisen aan een architectuur tool

De benodigde functionaliteit voor de ondersteuning van architecten door een architectuur tool kan ingedeeld worden in een aantal hoofdcategorieën (Zie ook Lankhorst):

- modellering en ontwerp;
- analyse van architecturen;
- visualisatie en publicatie;
- opslag en beheer.

Voor modellering en ontwerp is natuurlijk een belangrijke eis, dat er een (of meer) coherente modelleertaal over alle domeinen van architectuur binnen het tool wordt toegepast. Om over dit aspect van de beoordeling van een architectuurtool een goed oordeel te kunnen vellen, is het goed om eerst in detail in te gaan op de definitie van architectuur; ofwel van wat een architectuur - en dus ook een architectuurtool - moet bevatten.

Definitie van architectuur: de ISO/IEC/IEEE standaard

De standaard ANSI/IEEE 1471-2000 is in het jaar 2000 gepubliceerd als resultaat van het werk van de IEEE Architecture Working Group in de periode 1995-2000. Het doel van deze groep was te komen tot:

- een verzameling van referenties, termen en concepten ten behoeve van architectuur en haar beschrijving;
- het vastleggen van best practices voor architectuurbeschrijvingen van software-intensieve systemen;
- te zorgen voor verdere ontwikkeling van dit architectuur denken.

In 2000 is de standaard voor het eerst gepubliceerd door de IEEE. Deze standaard is breed geaccepteerd in de architectuur wereld en wordt door veel methoden onderschreven of als uitgangspunt gebruikt. In 2001 is de IEEE standaard door ANSI geaccepteerd als standaard voor de USA. In 2006 adopteerde de International Standardization Organization (ISO) IEEE 1471 als een internationale standaard.

Inmiddels is een nieuwe versie van deze standaard gepubliceerd onder de naam ISO/IEC/IEEE 42010:2011 - Systems and software engineering—Architecture description. De standaard is niet gratis, maar kan online worden aangeschaft.

De standaard wordt door IEEE en ISO gepubliceerd als:

- ANSI/IEEE Std 1471-2000, Recommended Practice for Architectural Description, en
- ISO/IEC 42010:2007 Systems and Software Engineering—Architectural Description.

IEEE en ISO werken gezamenlijk aan een revisie van de standaard, gebaseerd op gebruikerservaringen van de afgelopen acht jaren.

Zoals het doel en de naam al aangeven is de standaard geen werkwijze maar een standaard voor het beschrijven van architecturen en bestaat het uit een reeks inhoudsvereisten bij het voorbereiden van architectuurbeschrijvingen. De standaard is bedoeld om in combinatie met bestaande architectuurmethodes of binnen andere architectuurraamwerken te worden toegepast. De standaard biedt aanknopingspunten voor het correct gebruik van architectuurbeschrijvingen in architectuurtrajecten door expliciet uit te gaan van belanghebbenden (stakeholders) en hun belangen (concerns).

Het oude metamodel van IEEE is net als de IEEE definitie zeer bekend geworden. De nieuwe versie van het model is uitgebreid met een aantal nieuwe concepten, waarin het voortschrijdende denken over architectuur is weergegeven en waardoor de standaard nu beter bruikbaar is in een architectuur context. Het oude model had nog iets teveel de associatie met systeemontwikkeling.

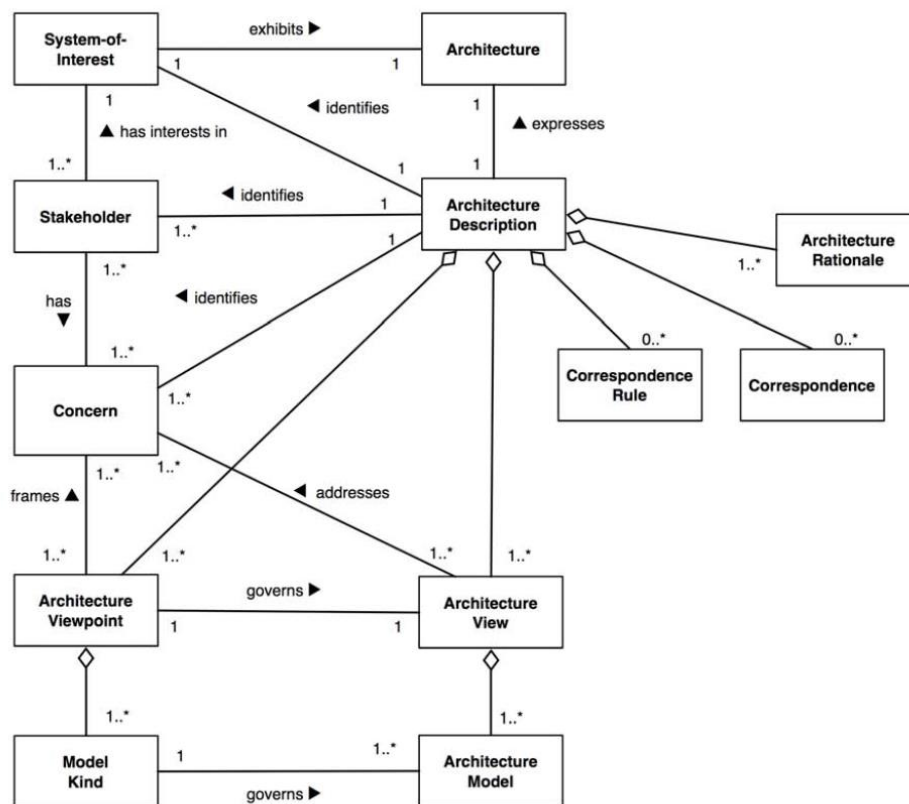
De definitie van architectuur in ISO/IEC/IEEE 42010:2011 luidt als volgt:

“The architecture (of a system) consists of the fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution”.

Ieder geordend geheel kan als systeem beschouwd worden. De architectuur van een systeem kan dus betrekking hebben op een organisatie(onderdeel) maar ook op (een onderdeel van) de informatievoorziening of een combinatie daarvan.

Deze nieuwe definitie van architectuur is wederom breed geaccepteerd in de internationale architectuurgemeenschap. De focus van de standaard ligt op de architectuurbeschrijving, dat wil zeggen de producten die gebruikt worden om de architectuur van een gekozen systeem te documenteren. De standaard is gebouwd rondom verschillende fundamentele begrippen, zoals belanghebbenden van het systeem (stakeholders), de belangen, die zij bij het systeem hebben (concerns), modellen en modeltypen, architectuurviews en viewpoints. Bovendien geeft de standaard aan in welke relatie de begrippen tot elkaar staan.

In figuur 1 zijn deze begrippen met hun onderlinge verbanden opgenomen.



NOTE 1 The figure uses the conventions for class diagrams defined in [ISO/IEC 19501].

Figuur 1 Het metamodel van de ISO/IEC/IEEE standaard

In het ISO/IEC/IEEE metamodel zijn de onderdelen van een architectuur weergegeven in hun onderlinge relaties.

Wanneer we instrumenten voor architectuur beoordelen, is het verstandig te stellen, dat dit tool in staat moet zijn beschrijvingen te genereren, die voldoen aan het ISO/IEC/IEEE metamodel voor architectuur. Anders gezegd: in een tool moeten alle onderdelen van het metamodel met hun onderlinge relaties terug ondersteund worden.

Met het tool moeten niet alleen modellen gemaakt kunnen worden, maar moeten die modellen ook views representeren, die de belangen (concerns) van de belanghebbenden (stakeholders) weergegeven. Architectuurbeschrijvingen moeten ook gerelateerd kunnen worden aan regels en principes (rationale).

Een coherente modelleertaal

Het kiezen van een geschikte tool voor architectuur moet voor architecten in de eerste plaats gaan over het kiezen van een geschikte gezamenlijke modelleertaal. Het inzetten van een tool heeft immers tot doel het bieden van samenhang en overzicht over de verschillende architectuur domeinen. Dat betekent, dat alle architecten in de verschillende domeinen ermee moeten kunnen en willen werken, anders schiet het inzetten van een tool zijn doel voorbij. Er moet dus tenminste overeenstemming zijn over de geschiktheid van de gebruikte modelleertaal voor de verschillende domeinen.

Zoals hiervoor al opgemerkt is, hanteren de verschillende domeinen ieder hun eigen, gespecialiseerde modelleerwijzen. Bijvoorbeeld procesontwerp en het ontwerpen van informatiestromen en applicaties kennen van oudsher hun eigen concepten en schematechnieken. Momenteel zijn op dit terrein er twee veelgebruikte open standaarden: BPMN (Business Process Model and Notation) en UML (Unified Modeling Language).

BPMN wordt niet alleen gebruikt om 'handmatige' processen mee te modelleren, maar ook voor het ontwerpen (soms zelfs ook runnen) van geautomatiseerde processen op een servicebus. Deze taal echter niet geschikt om architectuur over verschillende domeinen mee te modelleren.

Dat ligt anders voor UML. Hoewel UML afkomstig is uit de hoek van de software ontwikkeling, bestaat het uit een groot aantal modellen en modelleerwijzen, die ook bij architectuur kunnen worden gebruikt. Bovendien is UML zich aan het ontwikkelen richting een versie. UML wordt dan ook door een aantal architectuur tools ondersteund. Zeker, wanneer de architectuur sterk gericht is op het ontwerpen van software, zoals bij software leveranciers, is het gebruik van een tool voor architectuur gebaseerd op UML niet ongewoon. Het voordeel is dan namelijk, dat de modellen binnen hetzelfde tool eenvoudig kunnen worden doorontwikkeld tot modellen voor ontwerp of bijvoorbeeld deployment schema's.

Een andere bekende modelleertaal is ArchiMate. Het project dat de modelleertaal voor enterprise-architectuur ArchiMate heeft opgeleverd, is gestart in juli 2002 en heeft geduurd tot december 2004. Bij de aanvang van het project kwamen verschillende bedrijven en kennisinstellingen samen onder leiding van het Telematica Instituut om de bij het communiceren met en over architectuur ondervonden problemen te bespreken. De deelnemende partijen waren ABN AMRO, Belastingdienst, Stichting Pensioenfonds ABP, Radboud Universiteit Nijmegen, Centrum voor Wiskunde en Informatica (CWI), het Leiden Institute for Advanced Computer Science (LIACS) en Ordina. Deze partijen besloten het genoemde project uit te voeren. Na ongeveer 25 manjaren werk was ArchiMate begin 2005 een feit.

In januari 2005 werd een begin gemaakt met het ArchiMate Forum (later omgedoopt tot ArchiMate Foundation), dat ertoe diende het draagvlak voor het gebruik te vergroten ofwel het vergaren van kritische massa en dat zich tot doel stelde ArchiMate te laten uitgroeien tot een uniforme standaardtaal voor het modelleren van enterprise-architecturen. In 2008 heeft dit geleid tot de oprichting van het ArchiMate Forum binnen The Open Group, waarmee internationale standaardisatie en erkenning een feit was.

Begin 2012 is versie 2.0 van de ArchiMate specificatie uitgebracht. Sinds 2005 is dit de eerste grote vernieuwing, wat de stabiliteit van de modelleertaal bevestigt. De nieuwe versie is dan ook bovenal een uitbreiding. Aan de bestaande onderdelen is weinig veranderd.

In Nederland is ArchiMate de de facto standaard taal voor het modelleren van architectuur. Dat ArchiMate door een aantal Nederlandse bedrijven is ontwikkeld en hier al jaren bekend is, zal daar zeker mee te maken hebben. ArchiMate wordt bijvoorbeeld veel gebruikt in referentie architecturen voor een bepaalde sector.

3. Gebruikerseisen aan een architectuur tool

Allereerst zijn er natuurlijk een aantal eisen met betrekking tot het gemak van de gebruikersinterface. Eenvoudige invoer, een geavanceerde grafische interface, eenvoudige manipulatie van modellen, etc. zijn natuurlijk vereist. Maar dat is niet specifiek voor architectuur tools. Wat wel specifiek is, is dat het mogelijk moet zijn de objecten te voorzien van allerlei eigenschappen. Voorbeelden daarvan zijn het vastleggen van de eigenaar bij een applicatie, of het vastleggen van de bronhouder bij een authentiek data object. Deze eigenschappen moeten niet alleen kunnen worden vastgelegd, er moeten ook views of visualisaties rond dergelijke eigenschappen kunnen worden gegenereerd.

Een andere belangrijke eis - zeker in het licht van de opmerking in de inleiding bij dit hoofdstuk, dat je niet te snel met een tool moet beginnen - is, dat je modellen, die met andere tools zijn gemaakt, ook Visio, moet kunnen inlezen. Er zijn immers vaak al allerlei modellen en overzichten gemaakt, gebruikt en goedgekeurd. Het succes van de tool wordt mede bepaald door de mogelijkheid bestaande architecturen eenvoudig in te kunnen lezen, zodat niet alles opnieuw gemaakt hoeft te worden. Er zal natuurlijk altijd wel extra werk gedaan moeten worden als een tool in gebruik genomen wordt, bijvoorbeeld op het gebied van het aanbrengen van relaties tussen de verschillende domeinen en het aanbrengen van consistente naamgeving.

De architectuur tool moet natuurlijk in staat zijn de verschillende gezichtspunten of views te genereren op basis van het onderliggende model. De tool moet dus functionaliteit bieden voor :

- het maken van selecties uit het model, bijvoorbeeld alleen de bedrijfsservices met de bijbehorende doelgroep, alleen de processen in hun samenhang, etc.;
- het kiezen voor andere visualisaties, bijvoorbeeld een actor niet als ArchiMate blokje, maar als poppetje;
- het instellen van filters op het model, bijvoorbeeld alleen de objecten, die direct samenhangen met een bepaalde applicatie;
- verschillende manieren om relaties te tonen, bijvoorbeeld als pijl tussen objecten, als kleur, of door objecten in elkaar te tekenen;
- het tonen van objecten, relaties en eigenschappen als tabellen, kleuren of landschappen;
- het kiezen van specifieke grafische vormgeving, zoals kleuren, lettertypen en formaten, symbolen, het inlezen van afbeeldingen, etc..

Bij het creëren van views voor belanghebbenden is het belangrijk te kijken naar:

- De rol, die de visualisatie speelt. Is het ter ondersteuning van de besluitvorming, is het bedoeld ter ondersteuning van het ontwerp of om nieuwe gebruikers te informeren?
- De mate van detail, die gewenst is. Moet het een globaal overzicht bieden, ook de samenhang van objecten weergeven of juist zoveel mogelijk detail?

Uiteraard moet het ook eenvoudig mogelijk zijn de gegenereerde views te publiceren. Output moet zowel geschikt zijn om in Word of PowerPoint op te nemen, als om op een intranet of internet site te publiceren.

In de vorige paragraaf is al aan de orde geweest, dat vanuit het tool eenvoudig allerlei analyses gemaakt kunnen worden, zowel op consistentie en coherentie als inhoudelijke. Als voorbeelden daarbij zijn de impact analyse en de gap analyse. Bij het maken van dergelijke analyses is het belangrijk, dat de tool de resultaten zowel in een grafisch model, als in overzichtstabellen of matrices kan presenteren.

Een hele belangrijke eis aan een tool, is dat het in staat moet zijn indirecte relaties af te leiden. Als een proces gebruik maakt van een applicatieservice, onderdeel van een applicatie, die geïnstalleerd is op een server, dan hebben het proces en de server een relatie, die door het tool moet kunnen worden afgeleid. Met andere woorden, de tool moet ook indirecte relaties kunnen tonen, bijvoorbeeld alle processen, die op een server draaien.

Ook het vergelijken van verschillende architecturen, bijvoorbeeld tussen twee plateaus, zoals in een gap analyse gebeurt, moet door een tool ondersteund worden.

Een tool kan eigenlijk alleen voldoen aan al deze vereisten als ze over een repository beschikt. Een repository is in feite een database, waarin alle objecten met hun eigenschappen en relaties zijn opgeslagen. Bijkomend groot voordeel van een repository is, dat ook onderhoud en beheer van de architectuur daardoor veel eenvoudiger wordt. Een architectuur tool moet namelijk ook mogelijkheden bieden voor versiebeheer op concepten, op views en op architecturen. Dit uit zich in de volgende vereisten:

- Als een object wordt verwijderd, dan moeten ook alle bijbehorende relaties verwijderd worden;
- Objecten uit de repository moeten ieder hun eigen versie en historie kunnen hebben, zodat bepaalde documenten waar nodig oudere versies van een object kunnen gebruiken.
- Objecten moeten voorzien kunnen worden van labels, zoals 'onderdeel huidige situatie' of in gebruik tot 1-1-2014' zodat op basis van deze labels ook views gemaakt kunnen worden.
- Objecten moeten onderscheiden kunnen worden in status, zoals: in ontwerp, in besluitvorming, in gebruik.

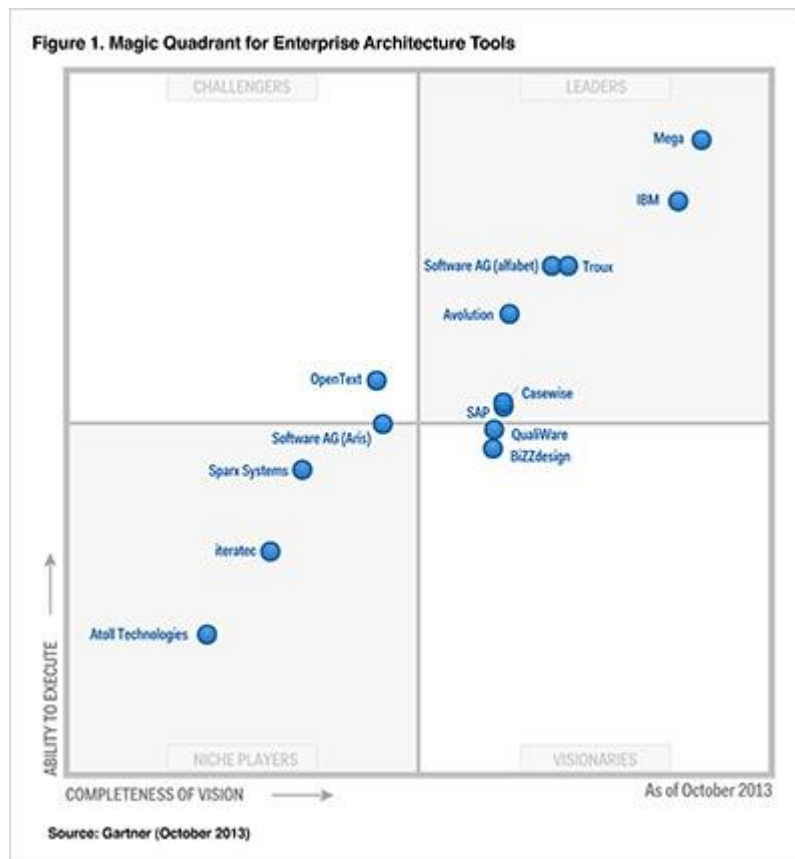
Uiteraard moeten de tool ook aan allerlei algemene vereisten voldoen. Denk daarbij aan het toekennen van (single sign on) gebruikersrechten, het beschikken over een locking mechanisme, zodat meerdere gebruikers tegelijkertijd aan dezelfde concepten kunnen werken, en vereisten met betrekking tot performance. Ook zullen technische vereisten met betrekking tot import en export geformuleerd moeten worden.

4. Tools selecteren

Om te komen tot een goede keuze voor een architectuur tool verdient het aanbeveling de binnen de organisatie normale werkwijze voor tool selectie te volgen op basis van de door de belanghebbenden opgestelde eisen en wensen. Zowel bij het opstellen van vereisten, als bij het opstellen van long lists en short lists zijn er instrumenten, die je kunt gebruiken bij het selecteren van architectuur tools.

Gartner magic quadrant voor architectuur tools

Gartner publiceert regelmatig een magic quadrant, ook voor enterprise architectuur tools. De meest recente is verschenen in oktober 2013.



Figuur 2 Gartner Magic Quadrant voor architectuur tools

Een Gartner quadrant zet twee aspecten tegenover elkaar:

1. De ability to execute, waarin factoren als de financiële positie van de leverancier, de reactie van de markt, productontwikkeling, verkoop kanalen en klantenomvang zijn samengevat versus
2. De completeness of vision, waarin het innovatievermogen van de leverancier, het leiden of volgen van de markt door de leverancier en de visie op de markt is weergegeven.

Deze twee assen leiden tot vier typen leveranciers/producten:

1. Leaders, die volwassen producten aanbieden, die voldoen aan de eisen van de markt. Zij demonstreren de visie die nodig is om hun marktpositie te behouden, terwijl de vraag verder evolueert.
2. Challengers, die weliswaar een goede positie hebben, maar geen sterke visie. Het kan dan bijvoorbeeld om een product gaan, dat wel een grote klantomvang heeft, maar aan het eind van zijn levenscyclus zit en daardoor minder goed en met minder visie onderhouden wordt.
3. Visionaries hebben wel een sterke visie op de markt, maar beschikken (nog) niet over het vermogen om daaraan te voldoen. Zij zijn soms in staat innovatieve producten te verkopen vooruitlopend op de algemene vraag. Zij introduceren vaak nieuwe technologieën. Of zij challengers of leaders worden hangt ervan af of klanten de nieuwe technologie accepteren.
4. Niche players doen het goed in een bepaald segment van de markt, of ze hebben niet het vermogen om het beter te doen dan andere leveranciers. Soms zijn het producten, die aan

het verdwijnen zijn. Hun producten kunnen perfect passen, maar het gevaar is dat het op de lange termijn geen goede keuze is.

Er zijn organisaties, die als regel hanteren, dat ze alleen producten in het leaders kwadrant van een Gartner magic quadrant aanschaffen.

Enterprise architecture tool selection guide

De enterprise architecture tool selection guide wordt opgesteld door het Institute for Enterprise Architecture Developments door Jaap Schekkerman en is momenteel toe aan versie 6.3. De gids beschrijft uitgebreid de vereisten, die aan een enterprise architectuur tool gesteld moeten worden en beoordelen tools daar ook op. Zij doet dit door een aantal groepen van functionaliteiten af te zetten tegen de typen professionals, die de tool gebruiken.

De functionaliteiten zijn gegroepeerd in een aantal inmiddels bekende onderwerpen:

- Methoden, modelleertaal en modellen;
- Ontwerp interface;
- Mogelijkheid tot automatisch genereren van modellen;
- Mogelijkheid om aan te passen en uit te breiden;
- Functionaliteit voor analyse;
- Repository;
- Deployment architectuur/technische vereisten;
- Kosten en leveranciersondersteuning;
- Resultaten, zoals views voor belanghebbenden, visualisaties, etc..

Op basis hiervan wordt een overzicht van de verschillende tools gegeven in termen van hun geschiktheid voor de verschillende gebruikersgroepen en hun voldoen aan de functionaliteiten.

Tenslotte bevat de gids ook een zeer uitgebreide lijst van vereisten en specificaties voor architectuur tools, die je kunt gebruiken om je eigen, specifieke vereisten mee op te stellen, of aan te toetsen.

5. Referenties

Literatuur

BizzDesign, *Quick Reference: Enterprise Architecture with TOGAF® and ArchiMate®*, www.bizzdesign.com

Lankhorst, Mark, *Enterprise Architecture at Work, Modelling, Communication and Analysis*, Springer Verlag, 2005

Schekkerman, J., *Enterprise Architecture Tool Selection Guide version 6.3*, Institute for Enterprise Architecture Developments, 2011, <http://www.enterprise-architecture.info>.

The Open Group, *ArchiMate® 2.0 specifications*, Van Haren Publishing, 2012.

Websites

The Open Group ArchiMate: <http://www.opengroup.org/archimate/>
Let op: ook voor gratis downloads moet je je eerst registreren.

Gartner analyses: <http://www.gartner.com/technology/home.jsp>

De ISO/IEC/IEEE standaard: <http://www.iso-architecture.org/42010/index.html>