

# Aantekeningen bij het SOA Manifest

Commentaar bij en toelichting op het SOA Manifest door Thomas Erl.

Service oriëntatie is een gedachtegoed dat richting geeft aan wat je doet. Service georiënteerde architectuur (SOA) is de technologie architectuur die het gevolg is van het toepassen van service oriëntatie principes.

Het was vanaf het begin duidelijk dat het een manifest zou worden over twee verschillende, maar nauw verwante onderwerpen: het service-georiënteerde architectuurmodel en service oriëntatie, het paradigma waarmee de architectuur gedefinieerd wordt. De vorm van dit manifest is afgekeken van 'the Agile Manifesto' waarbij de inhoud is beperkt tot bondige verklaringen die uitdrukking geven aan doelstellingen en waarden en aan de richtlijnen ter realisatie hiervan. Het manifest is daardoor geen specificatie, referentiemodel of white paper, en niet bedoeld om feitelijke definities in uit te werken. We hebben daarom besloten om dit voorwoord toe te voegen om toe te lichten hoe en waarom bepaalde termen gebruikt worden in andere delen van het manifest.

We passen service oriëntatie toe...

Het service oriëntatie gedachtegoed kan het best worden beschouwd als een manier om een bepaalde beoogde situatie te bereiken. Die beoogde situatie wordt nader gedefinieerd door een set strategische doelen en voordelen. Als we service oriëntatie toepassen creëren we software programma's en een technologie architectuur die helpen om die gewenste situatie te bereiken. Zo kunnen we ook bepalen of een technologie architectuur service georiënteerd is.

...om organisaties te helpen systematisch duurzame business waarde te leveren, waarbij de wendbaarheid toeneemt en de kosten dalen ...

Deze voortzetting van het voorwoord benadrukt de meest prominente en voor de hand liggende strategische voordelen van service-georiënteerde automatisering. Begrip van deze voordelen helpt bij het inzichtelijk maken van de eerder genoemde beoogde situatie die we door het toepassen van service oriëntatie willen bereiken.

Met de wendbaarheid van een bedrijf wordt het vermogen van een organisatie om te reageren op verandering bedoeld. Hoe makkelijker en effectiever een organisatie kan reageren op veranderingen in de business, des te efficiënter en succesvoller zal het zich kunnen aanpassen aan de effecten van veranderingen (om daarnaast ook te kunnen profiteren van eventuele voordelen die de verandering met zich

meebrengt).

Service oriëntatie positioneert services als IT bedrijfsmiddelen (assets) waarvan verwacht wordt dat ze herhaaldelijk waarde leveren die de initieel noodzakelijke investering op den duur duidelijk overtreft. Kosteneffectiviteit hangt primair af van deze verwachte 'Return On Investment'. Op diverse manieren gaat een grotere kosteneffectiviteit samen met een toegenomen wendbaarheid; als er meer mogelijkheden zijn om bestaande services te hergebruiken, zal dat vaak leiden tot lagere kosten bij de bouw van nieuwe oplossingen.

"Duurzame" business waarde verwijst naar de lange termijn doelstellingen van service oriëntatie waarbij software programma's zich manifesteren als services die de flexibiliteit bezitten om telkens opnieuw te worden ingezet als onderdeel (bouwblok) van nieuwe oplossingen, én kunnen worden aangepast aan de immer veranderende business eisen.

... in overeenstemming met de veranderende behoeften van de business.

Deze laatste woorden van het voorwoord vormen de sleutel tot het begrip van de onderliggende filosofie van service-georiënteerde automatisering. De noodzaak om business veranderingen voordurend mogelijk te maken vormt het fundament van service oriëntatie en wordt beschouwd als een fundamentele, overkoepelende strategische doelstelling.

Vanuit onze praktijkervaring zijn we tot de volgende prioritering gekomen:

De hierna volgende stellingen positioneren een set van kernwaarden, waarbij het belang van de ene waarde telkens hoger wordt gesteld dan dat van een andere waarde. Het doel van deze waarde vergelijking is het benoemen van en richting geven aan de moeilijke afwegingen die regelmatig moeten worden gemaakt om de strategische doelen en voordelen van service-georiënteerde automatisering op een consistente manier te behalen.

Business waarde gaat boven technologie strategie

Zoals eerder aangegeven vormt de noodzaak om business veranderingen mogelijk te maken een overkoepelend strategisch doel. Daarom is de fundamentele kwaliteit van SOA en van elk software programma, oplossing en ecosysteem dat voortkomt uit de toepassing van service oriëntatie dat zij 'business-driven' zijn. Het is niet de technologie die de richting van de business te bepaalt, maar het is de business visie die bepaalt hoe en welke technologie wordt gebruikt. Deze afweging kan verreikende gevolgen hebben in de IT gemeenschap binnen een bedrijf.

De afweging brengt veranderingen met zich mee in bijna alle fasen van de IT voortbrengingscyclus, vanaf de planning en de budgettering van geautomatiseerde oplossingen, tot hoe we deze bouwen en beheren. Alle andere kernwaarden en principes in het manifest ondersteunen op enigerlei wijze de verwezenlijking van deze waarde.

Strategische doelen gaan boven projectgebonden voordelen

In het verleden richtten veel IT projecten zich slechts op de bouw van applicaties die speciaal werden ontworpen om aan de bedrijfsproces behoeften van dat moment te voldoen. Dit voorzag in de directe (tactische) behoefte, maar naarmate er meer van deze applicaties-voor-één-doel werden opgeleverd resulteerde dit in een IT landschap vol eilandjes van bedrijfsregels en bedrijfsgegevens, die we applicatie silo's noemen. Zodra nieuwe business behoeften ontstonden, werden nieuwe silo's gecreëerd of ontstonden er integratie kanalen tussen de silo's. Aangezien zich steeds opnieuw wijzigingen in de business voordeden, werden de integratie kanalen steeds groter, moesten steeds meer silo's worden gemaakt, en uiteindelijk werd het IT landschap een labyrint en steeds moeilijker, tegen meer kosten en trager aan te passen.

In veel opzichten onstond service oriëntatie als reactie op deze problemen. Het is een gedachtegoed dat een alternatief biedt voor project specifieke, op silo's gebaseerde, integratie vergende applicatie ontwikkeling door onvoorwaardelijk voor de lange termijn (strategische) doelstellingen te kiezen. De beoogde situatie die wordt nagestreefd door service oriëntatie kent geen applicatie silo's. En ook al bestaan er legacy systemen en applicatie silo's, daar waar service oriëntatie wordt geadopteerd is het doel deze zoveel mogelijk te harmoniseren (zodat ze hun 'partijtje mee kunnen blazen' door services te leveren).

Intrinsiek vermogen tot samenwerking gaat boven maatwerk integraties

Om gegevens te kunnen uitwisselen moeten software programma's met elkaar kunnen samenwerken. Als software programma's niet zijn ontworpen om op elkaar aan te sluiten, is het niet waarschijnlijk dat ze zonder meer kunnen samenwerken. Om twee niet op elkaar aansluitende software programma's te laten samenwerken is er integratie nodig. Integratie is daarmee de inspanning die noodzakelijk is om samenwerking tussen verschillende software programma's mogelijk te maken.

Hoewel maatwerk integratie vaak nodig is kan het kostbaar en tijdrovend zijn en leiden tot kwetsbare architecturen die lastig zijn aan te passen. Een van de doelstellingen van service oriëntatie is ervoor te zorgen dat doelgerichte integratie inspanningen nauwelijks meer nodig zijn door software programma's (binnen een gegeven domein) zo te ontwikkelen dat ze van nature onderling kunnen samenwerken. Deze eigenschap wordt het intrinsiek vermogen tot samenwerking (intrinsic interoperability) genoemd. Het service oriëntatie gedachtegoed omvat een set specifieke ontwerp principes die erop gericht zijn om dit intrinsiek vermogen tot samenwerking te bereiken op verschillende niveaus.

Het intrinsiek vermogen tot samenwerking, als kenmerk van software programma's binnen een gegeven domein, is een belangrijke factor bij de realisatie van strategische voordelen als toename van de kosteneffectiviteit en wendbaarheid.

Herbruikbare services gaan boven doelspecifieke oplossingen

Zoals eerder uitgelegd staat service oriëntatie voor een ontwerp aanpak die bestaat uit een aantal ontwerp principes. Indien deze afdoende worden toegepast zullen ze ertoe leiden dat het software programma wordt vormgegeven als een eenheid service-georiënteerde bedrijfsregels, die met recht kan worden aangeduid als een service.

Services zijn uitgerust met concrete eigenschappen (die er bijvoorbeeld voor zorgen dat ze samen kunnen werken zonder dat integratie inspanning nodig is) die direct het eerder beschreven doel ondersteunen. Eén van die eigenschappen is het bundelen van breder toepasbare bedrijfsregels, zodat deze kunnen worden gedeeld en gebruikt bij het automatiseren van verschillende bedrijfsprocessen.

Een herbruikbare service manifesteert zich als een IT bedrijfsmiddel dat een duurzame business waarde vertegenwoordigt, terwijl het de kosten van en inspanning voor het leveren van nieuwe automatiseringsoplossingen terugdringt. Hoewel traditionele applicaties-voor-één-doel ten behoeve van tactische business doelstellingen zeker waardevol zijn, levert het gebruik van herbruikbare generieke services méér waarde op bij het realiseren van de strategische doelen van service-georiënteerde automatisering (waaronder opnieuw een toename van kosteneffectiviteit en grotere wendbaarheid).

### Flexibiliteit gaat boven optimalisatie

Dit is wellicht de breedste van de prioriteit stellingen, en kan het best worden beschouwd als een richtinggevende filosofie bij het maken van afwegingen tijdens het produceren en uitbouwen van individuele services en service portfolio's.

Optimalisatie verwijst primair naar het behalen van tactische winst door het tunen van het applicatie ontwerp of versnellen van oplevering om aan korte termijn behoeften te voldoen. Er is niets verkeerd aan optimalisatie, behalve dan dat het kan leiden tot de eerder genoemde IT landschappen met applicatie silo's als in de overwegingen flexibiliteit niet in de juiste mate wordt meegewogen.

Flexibiliteit als karakteristiek gaat bijvoorbeeld verder dan het vermogen van services om effectief (en intrinsiek) gegevens uit te wisselen. Om snel te kunnen inspelen op de alsmaar veranderende business behoeften moeten services ook beschikken over de flexibiliteit om te worden gecombineerd en geassembleerd tot samengestelde oplossingen. In tegenstelling tot traditionele gedistribueerde applicaties die veelal nogal statisch van aard zijn ondanks het feit dat zij uit verschillende componenten bestaan, moeten service composities worden ontworpen met een niveau van 'aangeboren' flexibiliteit dat continue verbetering mogelijk maakt. Dit betekent dat wanneer een bestaand bedrijfsproces verandert of als een nieuw business proces wordt geïntroduceerd, we in staat moeten zijn om services toe te voegen, te verwijderen of uit te breiden binnen de bestaande compositie met een minimale (integratie) inspanning. Om deze reden is de eigenschap om gemakkelijk gebruikt te kunnen worden in een compositie het doel van een van de belangrijkste ontwerp principes van service oriëntatie (service composability).

### Evolutionaire verfijning gaat boven het nastreven van initiële perfectie

Er is een veel voorkomend punt van verwarring als het gaat om de term wendbaarheid (agility) in relatie tot service oriëntatie. Sommige ontwerp methodes propageren een snelle, regelmatige oplevering van software programma's voor direct gebruik. Dit kan worden gezien als 'tactische wendbaarheid', omdat de focus ligt bij de tactische, korte termijn voordelen. Service oriëntatie bepleit het bereiken van wendbaarheid op een organisatie of business niveau met de bedoeling om de organisatie als geheel beter te laten reageren op veranderingen. Deze vorm van

wendbaarheid kan worden gezien als 'strategische wendbaarheid', omdat de nadruk ligt op een lange tijdspanne waarin met elk opgeleverd software programma toegewerkt wordt naar de beoogde situatie die wendbaarheid beschouwt als een strategische (lange termijn) waarde.

Om wendbaarheid van de organisatie binnen de IT gemeenschap van het bedrijf mogelijk te maken, moet het zich samen met de business ontwikkelen. We kunnen meestal niet voorspellen hoe een bedrijf zich zal moeten ontwikkelen door de jaren heen, en daarom is het niet mogelijk in één keer de perfecte service te bouwen. Tegelijkertijd is er vaak al veel bedrijfskennis opgebouwd binnen een organisatie die kan worden benut tijdens de analyse en modelleer fase van een SOA project.

Deze informatie kan ons, samen met de principes van service oriëntatie en toepassing van bewezen methodologieën, helpen een stel services te definiëren die het hedendaagse reilen en zeilen van de business goed ondersteunen en voldoende flexibel zijn om zich aan te passen aan toekomstige veranderingen.

Hoewel wij de onderwerpen rechts waarderen, hechten we méér waarde aan de onderwerpen links.

Door het bestuderen van bovenstaande prioritering, verkrijgen we inzicht in wat service oriëntatie onderscheidt van andere paradigma's. Met dit inzicht kunnen IT'ers op verschillende manieren hun voordeel doen. Het kan bijvoorbeeld helpen bij het formuleren van fundamentele criteria die kunnen worden gebruikt om te bepalen of en in welke mate service oriëntatie aansluit bij een bepaalde organisatie of de IT gemeenschap binnen dat bedrijf. Daarnaast kan het helpen bij het vaststellen in welke mate service oriëntatie kan of zou moeten worden geadopteerd.

De waardering van de kernwaarden kan ook inzichtelijk maken hoe groot de uitdaging is om met succes SOA projecten binnen bepaalde omgevingen uit te rollen. Zo zouden sommige kernwaarden lijnrecht tegenover diepgewortelde overtuigingen en voorkeuren kunnen staan. Dan moeten de voordelen van service oriëntatie worden afgewogen tegen de inspanning en de impact die de adoptie van service oriëntatie zou hebben (niet alleen ten aanzien van de technologie, maar ook ten aanzien van de organisatie en de IT cultuur).

De navolgende richtinggevende principes worden aangedragen om te helpen bij veel van dit soort uitdagingen.

Wij hanteren deze principes:

Tot dusver heeft het manifest de overkoepelende visie neergezet inclusief kernwaarden die bij die visie horen. Het vervolg van de verklaring bestaat uit een aantal principes die worden aangedragen als richtlijnen die helpen om aan deze waarden vast te houden en de visie te realiseren.

Het is belangrijk je te realiseren dat deze principes specifiek bij dit manifest horen. Er bestaat al een aparte set geaccepteerde ontwerp principes gericht op het service oriëntatie gedachtegoed en er zijn nog veel meer praktijkvoorbeelden en patronen specifiek voor service oriëntatie en service-georiënteerde architectuur beschreven.

Respecteer de sociale en gezagsstructuur van een organisatie.

Eén van de grootste valkuilen bij SOA adoptie is om het te benaderen als een technologie initiatief. Zo'n aanpak mislukt bijna altijd, omdat we niet voorbereid zijn op de onvermijdelijke gevolgen voor de organisatie.

Het invoeren van service oriëntatie gaat in de kern over het veranderen van de manier waarop we de business automatiseren. Dit betekent dat we, hoe we ook denken deze verandering door te voeren, altijd moeten beginnen met het doorgronden en begrijpen van de organisatie, haar structuur, doelen en cultuur.

Het overschakelen naar service oriëntatie is vooral een zaak die mensen raakt. Het vraagt steun van de bedrijfsleiding en vereist dat de IT cultuur wijzigt naar strategisch denkend en gemeenschapsgericht. We moeten dit niveau van veranderingen in het bedrijf onderkennen en het in de plannen meenemen om de noodzakelijke lange termijn steun te krijgen om het doel van service oriëntatie te bereiken.

Dit soort overwegingen helpt niet alleen om te bepalen hoe een SOA initiatief door te voeren, maar ze helpen ook bij het kiezen van de meest geschikte reikwijdte en aanpak voor SOA adoptie.

Onderken dat SOA uiteindelijk verandering vereist op vele niveaus.

Er is een (Amerikaans) gezegde dat luidt: "Succes bereik je door klaar te staan om van kansen te profiteren". Misschien is de belangrijkste les die we kunnen leren van de SOA projecten die tot dusver zijn uitgevoerd dat we de hoeveelheid en de omvang van veranderingen die de invoering van service oriëntatie met zich meebrengt goed moeten begrijpen, plannen en voorbereiden. Hieronder volgen enkele voorbeelden.

Service oriëntatie verandert hoe we geautomatiseerde oplossingen maken, door software programma's te positioneren als IT bedrijfsmiddelen (assets) met een duurzame (lange termijn) waarde voor de business. Er is een investering voor nodig om een omgeving te creëren die uit zulke bedrijfsmiddelen is opgebouwd, en er is een voortdurende inspanning nodig om de waarde ervan in stand te houden en uit te bouwen. Dus zullen direct vanaf het begin veranderingen noodzakelijk zijn binnen de IT gemeenschap van het bedrijf in hoe we systemen bekostigen, op waarde schatten en onderhouden.

Omdat service oriëntatie services introduceert die worden gepositioneerd als bedrijfsmiddelen (enterprise resources), zullen er verder veranderingen nodig zijn in 'eigenaarschap' van verschillende delen van systemen en hoe we het ontwerp en gebruik ervan regelen, om maar te zwijgen van de infrastructuur die nodig is om continue schaalbaarheid en betrouwbaarheid te garanderen.

De mate waarin SOA geïmplementeerd wordt kan variëren. Houd de inspanningen beheersbaar en binnen betekenisvolle kaders.

Het is een wijdverbreide misvatting dat om de strategische doelen van service-georiënteerde software ontwikkeling te bereiken, service oriëntatie in de gehele onderneming moet worden ingevoerd. Wat zoveel inhoudt als het invoeren en bewaken van standaarden voor ontwerp en technologie binnen de hele IT gemeenschap van de onderneming zodat er een ondernemingsbreed portfolio van services ontstaat die intrinsiek met elkaar kunnen samenwerken. Hoewel er niets

mis is met zo'n ideaal, is dit voor veel organisaties niet realistisch, vooral niet voor organisaties met een grote IT gemeenschap.

De meest geschikte reikwijdte moet voor elke poging tot invoeren van SOA worden vastgesteld op basis van planning en analyse in samenhang met pragmatische afweging van de eerder genoemde impact op organisatie structuren en gezagsverhoudingen, en de benodigde cultuurveranderingen.

Dit soort factoren helpen bij het vaststellen van een reikwijdte die beheersbaar is. Daarnaast moet, net als voor elke vernieuwing die de IT gemeenschap verder moet brengen naar een gewenst strategisch niveau, de reikwijdte wel betekenisvol zijn. Met andere woorden, het moet een betekenisvolle doorsnede van de organisatie betreffen, zodat verzamelingen gerelateerde, zinvolle services kunnen ontstaan binnen vooraf bepaalde grenzen. Kortom we willen 'service continenten' maken, en niet de gevreesde 'service eilandjes'.

Dit concept, waarin per domein van dezelfde onderneming afzonderlijke bestuurd service portfolio's worden opgebouwd, helpt om de risico's te ondervangen die geassocieerd worden met 'big bang' SOA projecten en beperkt tevens de impact van zowel de organisatorische als technologische veranderingen (omdat de impact beperkt kan blijven tot een beperktere, beheersbare reikwijdte). Zo'n aanpak maakt het ook mogelijk om een gefaseerde invoering te plannen, waarbij in verloop van tijd steeds voor één domein een service portfolio wordt opgebouwd.

Producten en standaarden alleen zullen niet leiden tot SOA, noch zullen ze het service oriëntatie gedachtegoed voor je implementeren.

Dit principe raakt twee verschillende, maar gerelateerde mythen. De eerste is dat je SOA kunt kopen door moderne technologie te introduceren, en de tweede is de aanname dat het adopteren van industrie standaarden (XML, WSDL, SCA en dergelijke) als vanzelf een service-georiënteerde technologie architectuur zal opleveren.

Leveranciers en standaardisatiegroepen hebben ons moderne service technologie gebaseerd op open frameworks en platformen geleverd. En diverse ontwikkelingen (van service virtualisatie tot cloud computing en grid computing) hebben het gemakkelijker gemaakt om geavanceerde en complexe service-georiënteerde oplossingen te maken. Echter, deze technologieën zijn geen van allen exclusief van toepassing op SOA. Je kunt evengoed een applicatie silo in de cloud bouwen in plaats van op je eigen servers.

Je kunt SOA niet kopen, simpelweg omdat om een service-georiënteerde technologie architectuur te bereiken, service oriëntatie met succes moet worden toegepast; en hiervoor is het weer nodig dat alles wat we ontwerpen en bouwen wordt gedreven door de unieke richting, visie, en behoeften van de business.

SOA kan worden gerealiseerd met behulp van een scala van technologieën en standaarden.

Service oriëntatie een gedachtegoed dat technologie en leverancier neutraal is. Service-georiënteerde applicaties kunnen worden gezien als een gespecialiseerde vorm van gedistribueerde applicaties. Service-georiënteerde oplossingen kunnen

daarom worden gemaakt met zo'n beetje elke denkbare technologie en industrie standaard die je kunt gebruiken voor het ontwikkelen van gedistribueerde systemen.

Hoewel sommige technologieën (met name die die gebaseerd zijn op industrie standaarden) het in zich hebben om meer te halen uit de toepassing van bepaalde service-georiënteerde ontwerp principes, vormen in feite de mogelijkheden om business behoeften te verwezenlijken de doorslaggevende factor in de keuze van de meeste geschikte technologie en industrie standaarden.

Stel een samenhangende set van bedrijfsbrede standaarden en normen en voorschriften vast, gebaseerd op industrie-, de facto, en community standaarden.

Industrie standaarden zijn leveranciers- en productonafhankelijke technologie specificaties voor het vaststellen van consistente basis karakteristieken (zoals transport, interface, bericht formaat, etc.) van de technologische architectuur. Echter, het gebruik van industrie standaarden garandeert op zichzelf niet dat services een intrinsiek vermogen tot samenwerking bevatten.

Om twee software modules moeiteloos met elkaar te laten samenwerken zijn aanvullende afspraken (zoals gegevensmodellen en gebruiksnormen (policies)) nodig. Dit is de reden waarom de IT gemeenschap van een onderneming ontwerpstandaarden moet vaststellen en afdwingen. Als het standaardiseren en reguleren van de standaardisatie van services faalt, zal dit negatieve gevolgen hebben voor de interoperabiliteit van services waarvan de realisatie van de vele strategische voordelen nu juist afhankelijk van is.

Dit principe propageert niet alleen het gebruik van bedrijfsbrede ontwerpstandaarden, het geeft ook aan dat op maat gemaakte ontwerpstandaarden zoveel mogelijk gebaseerd moeten zijn op en gebruik moeten maken van standaarden die al in gebruik zijn in de industrie en de gemeenschap.

Streef naar uniformiteit aan de buitenkant, sta tegelijkertijd diversiteit toe aan de binnenkant.

Een federatie kan worden gedefinieerd als de vereniging van een aantal verschillende entiteiten. Terwijl elke entiteit onafhankelijk bestuurd wordt aan de binnenkant, houden zij zich allemaal aan één uniforme, gemeenschappelijk afgesproken, buitenkant.

Een fundamenteel onderdeel van de service-georiënteerde architectuur is de introductie van een uniforme laag aan de buitenkant die de service implementatie details afschermt en tegelijkertijd de individuele services van een domein op een uniforme manier weergeeft. Dit vereist over het algemeen dat er uniformiteit wordt nagestreefd op basis van een combinatie van industrie en ontwerp standaarden. Het consistent doorvoeren van deze uniformiteit voor alle services is de sleutel tot het realiseren van het intrinsiek vermogen tot samenwerken.

Een uniforme laag aan de buitenkant vergroot de keuzemogelijkheden van platform- en productleveranciers. Het kan bijvoorbeeld nodig zijn dat één service op een volledig ander platform draait dan een andere service. Zolang deze services op dezelfde, uniforme manier kunnen worden gebruikt, kan de besturing van de respectievelijke implementaties volkomen onafhankelijk van elkaar worden



uitgevoerd. Behalve dat services gebouwd kunnen worden middels verschillende implementatie media (zoals EJB, .NET, SOAP, REST, etc.), benadrukt dit dat ook verschillende tussenliggende platformen ('middleware') en technologieën desgewenst naast elkaar kunnen worden gebruikt.

Hierbij dient wel aangetekend te worden dat deze vorm van diversiteit zijn prijs heeft. Dit principe propageert diversiteit niet - het geeft slechts aan dat diversificatie mogelijk moet zijn als het gerechtvaardigd is, zodat 'best of breed' technologieën en platformen kunnen worden benut om de invulling van business behoeften te optimaliseren.

Identificeer services door business en technologie belanghebbenden te laten samenwerken.

Als de technologische oplossingen door business doelen geïnspireerd moeten zijn, zal de technologie zelf ook in sync moeten zijn met de business. Daarom is een ander doel van service-georiënteerde automatisering het afstemmen van technologie en business. Deze afstemming wordt initieel bereikt in de analyse en modelleringsfase die normaal gesproken voorafgaat aan de uiteindelijke service ontwikkeling en oplevering.

Het belangrijkste ingrediënt bij het uitvoeren van een service-georiënteerde analyse is een samenwerking tussen business en technologie deskundigen bij het identificeren en definiëren van de kandidaat services. Zo kunnen de business deskundigen bijvoorbeeld nauwkeurig de functionele context definiëren die past bij business specifieke services, terwijl de technologie experts er vanuit pragmatisch oogpunt voor kunnen zorgen dat de granulariteit en de definitie van de concept-services realistisch blijft met het oog op de omgeving waarin ze uiteindelijk worden gerealiseerd.

Maximaliseer service gebruik door rekening te houden met zowel huidig als toekomstig gebruik.

De omvang van een SOA project kan bedrijfsbreed zijn of gelimiteerd tot een domein binnen het bedrijf. Wat de reikwijdte ook is, er moeten vooraf gedefinieerde kaders worden vastgesteld die een portfolio van services omvatten, welke conceptueel gemodelleerd dienen te worden, voordat ze ontwikkeld kunnen worden. Door de verschillende services te modelleren in relatie tot elkaar, ontstaat feitelijk een blauwdruk van de services die we uiteindelijk zullen bouwen. Deze modelleringsexercitie is een essentiële stap bij het identificeren en definiëren van een portfolio van services welke gebruikt kunnen worden in verschillende oplossingen.

Er zijn verschillende methodologieën en benaderingen die gebruikt kunnen worden voor het uitvoeren van de diverse fasen van een service-georiënteerde analyse. Ze hebben echter allemaal gemeen dat de functionele grenzen van services worden genormaliseerd om redundantie te voorkomen. Echter het feit dat een service genormaliseerd is, maakt hem niet als vanzelf goed herbruikbaar. Meerdere factoren spelen hierbij een rol, zoals granulariteit, autonomie, state management, schaalbaarheid, hoe gemakkelijk een service gebruikt kan worden als onderdeel van een compositie (composability) en de mate waarin de service logica voldoende generiek is, zodat het in de praktijk hergebruikt kan worden.

Dit soort overwegingen leidt, gebaseerd op business en technologie expertise, tot de definitie van services die de huidige gebruiksbehoeften weerspiegelen, en die de flexibiliteit bieden om zich aan te passen in de toekomst.

Verifieer dat de services voldoen aan de behoeften en doelstellingen van het bedrijf.

Zoals met zoveel dingen is het goed mogelijk services verkeerd te gebruiken. Bij het groeien van het service portfolio, zal het gebruik van individuele services en de mate waarin ze (nog) voldoen aan de business behoeften, regelmatig moeten worden geverifieerd en gemeten. Moderne tools bieden diverse mogelijkheden om het gebruik van software services inzichtelijk te maken, maar er spelen ook aspecten een rol die niet zonder meer meetbaar als je wilt vaststellen of een service niet alleen gebruikt wordt omdat hij toevallig beschikbaar was, maar dat het gebruik ook daadwerkelijk voldoet aan de beoogde business doelstellingen.

Dit speelt vooral bij herbruikbare services waar meerdere partijen afhankelijk van zijn. Niet alleen is hiervoor een goede infrastructuur nodig die schaalbaarheid en betrouwbaarheid van de service garandeert voor alle oplossingen die er gebruik van maken, daarnaast zullen ze ook zorgvuldig moeten worden ontworpen (en later eventueel aangepast), om te borgen dat de functionele context van de service geen geweld wordt aangedaan.

Pas services en hoe ze georganiseerd zijn aan op basis van daadwerkelijk gebruik.

Dit principe sluit direct aan bij het "Evolutionaire verfijning gaat boven het nastreven van initiële perfectie" waarde oordeel en ook bij het algemene doel om business en technologie op elkaar afgestemd te houden.

We kunnen niet vertrouwen op giswerk als het aankomt op het bepalen van de granulariteit van de service, het reeks van functies die de services moeten uitvoeren of hoe services moeten worden georganiseerd binnen een compositie. Gebaseerd op de analyse die wie initieel hebben kunnen uitvoeren (vluchtig danwel diepgaand), krijgt een bepaalde service een gedefinieerde functionele context en één of meer functionele mogelijkheden die waarschijnlijk in een of meerdere service composities zal worden gebruikt.

Op het moment dat de business behoeften en omstandigheden veranderen, zal de service wellicht worden vergroot, uitgebreid, verbouwd of helemaal vervangen. Service- georiënteerde ontwerp principes zorgen voor ingebouwde flexibiliteit in service architecturen zodat, net als software programma's, services bestand zijn tegen en voorbereid zijn op wijzigingen in reactie op het daadwerkelijke gebruik

Scheid de verschillende aspecten van het systeem op basis van de snelheid waarmee ze veranderen.

Wat monolitische en op silo's gebaseerde systemen inflexibel maakt, is dat een verandering een significante impact kan hebben op het huidige gebruik. Dat is de reden waarom het vaak eenvoudiger is om een nieuwe applicatie silo te bouwen dan de bestaande uit te breiden of te vergroten.

De rationale achter het scheiden van verantwoordelijkheden (separation of

concerns, een algemeen bekende ontwerp principe binnen software engineering) is dat een groter probleem eenvoudiger kan worden opgelost door het op te splitsen in een set van kleinere problemen of verantwoordelijkheden. Wanneer service oriëntatie wordt gecombineerd met separation of concerns, bouwen we eenheden van logica met hun eigen individuele verantwoordelijkheden, waarbij het mogelijk is de eenheden te aggregeren tot een compositie dat een grotere verantwoordelijkheid invult, waarbij het tevens mogelijk is de eenheden te aggregeren in andere configuraties, zodat ook andere problemen opgelost en verantwoordelijkheden geadresseerd kunnen worden.

Naast het gunstige effect dat dit heeft op de herbruikbaarheid van services, brengt deze benadering verschillende abstractielagen die de impact van wijzigingen beperkt voor systemen samengesteld uit services. Deze vorm van abstractie kan bestaan op verschillende niveaus. Als bijvoorbeeld legacy systemen die één bepaalde service implementeren vervangen moeten worden, kan de impact voor oplossingen die de service gebruiken beperkt blijven zolang de service zijn originele 'adres' en functionele gedrag kan behouden.

Een ander voorbeeld is de scheiding van agnostische en niet-agnostische logica. Het eerste type logica heeft een grote kans om hergebruikt te worden zolang ze voor meerdere doeleinden geschikt is en niet te vaak verandert. Niet-agnostische logica daarentegen is echter typerend voor specifieke delen van het bovenliggende bedrijfsproces, die vaker veranderen. Het scheiden van deze twee soorten logica in verschillende service lagen introduceert verdere abstractie, die service hergebruik mogelijk maakt terwijl het services en elke oplossing die ze gebruikt afschermt van de gevolgen van verandering.

Beperk impliciete afhankelijkheden en publiceer alle externe afhankelijkheden met als doel robuustheid te vergroten en de impact van verandering te verkleinen.

Een van de meest bekende service-georiënteerde ontwerp principes is service loose coupling. Hoe een service architectuur intern gestructureerd is en hoe services zich verhouden tot de programma's die er gebruik van maken (waaronder zich andere, samengestelde services kunnen bevinden) - het zijn allemaal afhankelijkheden gevormd door de individuele bewegende delen die onderdeel uitmaken van de service architectuur.

Abstractielagen maken evolutionaire verandering eenvoudiger door de consequenties van de verandering te concentreren in gecontroleerde gebieden. Zo kunnen bijvoorbeeld service façades binnen service architecturen worden gebruikt om delen van de implementatie te abstraheren, waarmee de reikwijdte van implementatie afhankelijkheden beperkt blijft.

Aan de andere kant moeten de gepubliceerde technische service contracten de afhankelijkheden weergeven die de service afnemers hebben moeten vormen om met de services te interacteren. Door de interne afhankelijkheden van een service, die bij een verandering deze technische contracten kunnen beïnvloeden, beperkt te houden, vermijden we dat de impact van deze wijzigingen doorwerkt naar de service afnemers.

Organiseer elke service rond een samenhangende en

onderhoudbare eenheid van functionaliteit, dit op elk abstractieniveau.

Elke service vereist een goed gedefinieerde functionele context die bepaalt welke bedrijfsregels wel of niet binnen de functionele grens van de service vallen. Het bepalen van de reikwijdte en de granulariteit van deze functionele service grenzen is een van de meest essentiële verantwoordelijkheden uit de service voortbrengingscyclus.

Services met een grove functionele granulariteit zijn soms niet flexibel genoeg om effectief te zijn, vooral niet als ze bedoeld zijn om vaker gebruikt te worden. Daarentegen kunnen services met een heel fijne granulariteit de infrastructuur hoog belasten doordat service composities uit een groter aantal compositie deelnemers zullen moeten bestaan. Het bepalen van de juiste balans van de functionele scope en granulariteit vereist een combinatie van business en technologie expertise en daarnaast begrip van hoe de services binnen een gegeven grens zich tot elkaar verhouden.

Veel van de richtinggevende principes uit dit manifest helpen bij deze afweging op een manier dat elke service als een IT bedrijfsmiddel kan worden gezien, een bedrijfsmiddel dat de IT gemeenschap binnen een onderneming dichterbij die beoogde situatie brengt, waarin de strategische voordelen van service-georiënteerde automatisering worden gerealiseerd.

Uiteindelijk zal het kunnen verschaffen van business waarde voor de echte wereld bepalend zijn voor het evolutionaire pad van elke eenheid van service-georiënteerde functionaliteit, van idee tot oplevering tot hergebruik.

*Dankbetuiging: Ik schreef de tekst van dit document in het weekend van 21-22 november 2009, voor het 'Next Generation SOA' boek dat toen nog in ontwikkeling was. Ik wil graag Prentice Hall bedanken dat zij toestemming hebben gegeven om deze tekst publiekelijk weer te geven op deze website, als toevoeging op het originele SOA Manifest. Ik zou ook graag de leden van de werkgroep en van de SOAPatterns.org gemeenschap willen bedanken voor de feedback en de hulp bij het schrijven van deze toevoeging. Veel leden van de werkgroep hebben zelf ook artikelen, blogs en papers over het SOA Manifest gepubliceerd, en ik moedig een ieder aan om ook die te lezen om je inzicht te verdiepen.*

- Thomas Erl

---

## Vertalers

Laurens Van Berge Henegouwen  
Arco Keijzer  
Johan Kumps  
Brian Lokhorst  
Edwin Smink

---

[Original SOA Manifesto](#) [PDF](#) [About the Translators](#)

Copyright © 2009-2011