

Business Engineering

Volwassen Business Rule Management leidt tot Business Engineering

Alcedo Coenen, Leo Hermans, Matthijs van Roosmalen en Silvie Spreeuwenberg

Hierbij bieden wij u de unieke gelegenheid één heel hoofdstuk uit het boek “Uw bedrijf geregeld met Business Rule Management”¹ te lezen. Dit boek is geschreven door Alcedo Coenen, Leo Hermans, Matthijs van Roosmalen en Silvie Spreeuwenberg. Leo Hermans overleed 25 juli 2009, en deze reproductie is een eerbetoon aan zijn bijdrage. Het hoofdstuk wat wij gekozen hebben brengt Leo’s gedachtegoed representatief tot uiting.

Leo was in ons schrijversteam een belangrijke motor en zorgde met zijn bijdragen en discussies ervoor dat we de zaken scherp kregen. Hij heeft zich altijd op een voor hem kenmerkende energieke wijze ingezet voor wat hij Business Engineering noemde, een nieuwe discipline die inzichten uit de artificiële intelligentie, Business Rule Management en procesmanagement combineert.

Natuurlijk borduurt hoofdstuk 5 van het boek voort op de inzichten die reeds eerder in het boek zijn opgenomen. Wij nodigen u dan ook uit het boek in zijn geheel te lezen.

Met dank aan SDU uitgevers die toestemming verleenden voor deze internet reproductie.

Zoals in hoofdstuk 4 is aangegeven, betekent een grotere mate van volwassenheid in *Business Rule Management (BRM)* automatisch een koppeling met *Business Process Management (BPM)*. Dit leidt tot een nieuwe discipline die *Business Engineering* wordt genoemd. Dit hoofdstuk zal beginnen met een uitleg over hoe *Business Engineering* voortvloeit uit de tekortkomingen van *Software Engineering*, om vervolgens in te gaan op de specifieke eigenschappen en voordelen van *Business Engineering*. Tenslotte wordt ingegaan op de organisatorische impact die *Business Engineering* heeft op het inrichten van de bedrijfspraktijk.

¹ Zie <http://www.sdu.nl/catalogus/9789012125956> voor meer gegevens

De noodzaak voor een nieuwe discipline

Business Rule Management staat niet op zichzelf, zoals eerder is aangegeven in de hoofdstukken 1 en 4. *BRM* maakt deel uit van een aantal inrichtingsactiviteiten die ervoor moeten zorgen dat de bedrijfspraktijk zodanig werkt dat de bedrijfsdoelen worden bereikt. *BRM* is daarmee een stuurmiddel dat zich concentreert op bedrijfsregels, en dat moet samenwerken met stuurmiddelen op andere aspecten zoals bedrijfsprocessen en bedrijfsvocabulaire. Vooral als er naar gestreefd wordt om de afstemming te verbeteren tussen strategie en beleid enerzijds en bedrijfspraktijk anderzijds, zoals tot uitdrukking komt in de volwassenheidsmodellen van hoofdstuk 4, is het noodzakelijk om *BRM* en *Business Process Management (BPM)* integraal in te richten. Dat betekent dat men bij het inrichten en verder ontwikkelen van *BRM* al gauw terecht komt in een nieuwe discipline die beide verenigt, en die *Business Engineering* genoemd kan worden. *Business engineering* is echter meer dan alleen een integratie van *BRM* en *BPM*, zoals uit het navolgende zal blijken.

Het dilemma van Software Engineering

De dienstverlenende sector, vooral financiële instellingen, zorginstellingen en overheidsinstellingen, is in navolging van de industrie bezig om zichzelf vergaand te digitaliseren. Dit houdt in dat het niet-cruciale en routinematige werk in de bedrijfspraktijk dat weinig waarde toevoegt, wordt uitbesteed aan partners of flexibel wordt geautomatiseerd en dat diensten via verschillende media direct worden aangeboden aan zichzelf bedienende klanten en partners. De mate van automatisering neemt hierdoor steeds meer toe. Het gaat om automatisering zowel van processtromen en samenwerking als van het nemen van beslissingen. Het gevolg van deze vergaande automatisering is dat software meer is geworden dan een hulpmiddel. Tot nu toe was het gebruikelijk dat software in de dienstverlenende sectoren niet veel meer deed dan het verwerken en opslaan van gegevens wat ten dienste stond aan de medewerker die de eigenlijke dienstverlening naar de klant uitvoert. Software neemt in toenemende mate de routinematige functies van de medewerker over. Software ondergaat daarmee een metamorfose van louter hulpmiddel tot cruciale *business service*. Parallel daaraan treedt er een verschuiving op van routinewerkers naar kenniswerkers. Terwijl het werk van de routinewerker door de software wordt overgenomen, wordt het werk van de kenniswerker des te belangrijker. Kenniswerkers blijven de moeilijkere eventueel exceptionele beslissingen nemen en moeten de bedrijfskennis in de informatiesystemen onderhouden.

De manier waarop software wordt ontwikkeld speelt daarbij parten. De gangbare methoden en technieken van *software engineering* komen in toenemende mate onder vuur te liggen. Ondanks vernieuwde methoden zoals *Agile Development*, *Extreme Programming* en *Scrum* zijn de methodieken voor softwareontwikkeling vooral succesvol in technische omgevingen en bij relatief eenvoudige oplossingen, maar kunnen nauwelijks nog voldoen aan de vereisten van snel veranderende dienstverlening. Zolang software als hulpmiddel wordt gebruikt door medewerkers van dienstverlenende organisaties, blijft de complexiteit van de software beperkt tot het verwerken van gegevens, en blijft de communicatie beperkt tot de medewerker, die altijd nog een handleiding kan lezen of zich kan wringen in de bochten van door technologen construeerde software.

De reden waarom *software engineering* minder goed past in de groeiende digitaliseringambities van de dienstverlenende sector ligt in het feit dat *software engineering* een technologiecentrisch paradigma heeft. Dat brengt hoofdzakelijk twee nadelen met zich mee, die hieronder nader worden toegelicht:

1. Grote afstand tussen bedoeling en resultaat;
2. Verborgene bedrijfsregels.

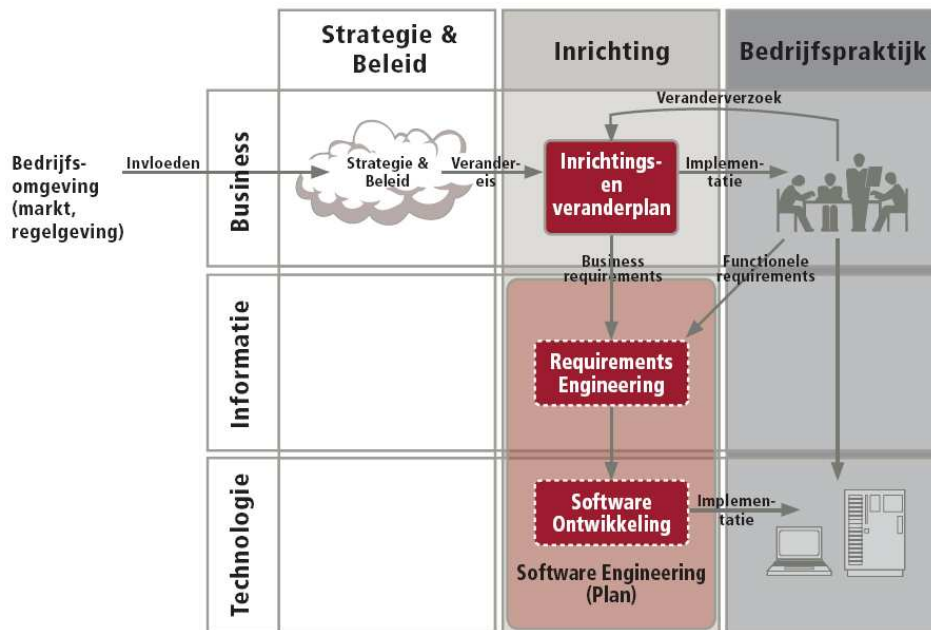
Grote afstand tussen bedoeling en resultaat

Zoals bekend bestaat er in de automatisering een groot risico op een mismatch tussen de bedoelingen (vereisten of *requirements*) en de uiteindelijk opgeleverde software. Nu bestaat er altijd een probleem bij het realiseren van een gewenst, nog niet-bestaand doel, omdat er een vertaalslag gemaakt moet worden tussen ideeën en de werkelijkheid. Dat geldt voor de

kunstenaar die een beeld wil maken, dat geldt evenzeer voor de manager die een bedrijfsdoel wil realiseren: dingen en mensen laten zich niet zomaar vormen tot een beoogd resultaat, daar is een proces voor nodig, en in dat proces kan de oorspronkelijke bedoeling gemakkelijk uit het oog worden verloren. De grote uitdaging bestaat er dan ook uit om de afstand tussen bedoeling, ontwerp en middelen zo klein mogelijk te houden. Bij de huidige methoden van *software engineering* is die afstand te groot.

De mismatch in de *software engineering* praktijk ontstaat door de volgende factoren:

- Dubbele aansturing.
Er is vaak sprake van een dubbele aansturing: enerzijds van de niet-technische bedrijfspraktijk (mensen en procedures), anderzijds van de technische bedrijfspraktijk (software als primair hulpmiddel).



Figuur 1: *De gebruikelijke Software engineering aanpak*

Figuur 1 laat in relatie tot het informatiemanagement model uit hoofdstuk 1, zien dat operationele organisatie en ICT traditioneel gescheiden worden aangestuurd bij het realiseren van veranderingen. De aanleiding voor verandering komt van buiten (verandereisen afkomstig van strategie en beleid) of van binnenuit (wijzigingsverzoek uit de bedrijfspraktijk), en zet de bestuurders aan tot verandering in de inrichting van de bedrijfspraktijk. Daartoe wordt in eerste instantie op businessniveau een inrichtings- en veranderplan gemaakt, dat zijn uitwerking moet krijgen in de bedrijfspraktijk zowel wat betreft organisatie als wat betreft ICT. Uitgaande van dit plan worden meestal afgeleide ontwikkelplannen voor de ICT-inrichting gemaakt, die snel hun eigen leven gaan leiden.

Aan het begin is er gedreven door ICT nog wel even contact met de business ten einde de vereisten (*requirements*) voor de ICT-inrichting te verzamelen. ICT schrikt er hierbij niet voor terug om zelf te bedenken wat de business nodig zou moeten hebben en omgekeerd rekent de business daar vaak gemakshalve op. Omdat ICT zijn eigen cultuur, methoden en technieken kent, die volstrekt vreemd zijn voor de business (en omgekeerd natuurlijk) gaan business en ICT vervolgens ieder voor een lange tijd huns weegs, wat het risico op een mismatch tussen de bedoelde bedrijfspraktijk en de gerealiseerde software enorm vergroot.

Een factor die deze problematiek vergroot is dat de onzekerheid over het beoogde resultaat groter wordt naarmate software minder als primair hulpmiddel en meer als *business service* moet gaan fungeren. De bedrijfsvoering is als systeem immers veel minder voorspelbaar en maakbaar dan een software hulpmiddel voor het managen van gegevens. Het automatiseren van complete selfservice processen voor klanten bijvoorbeeld moet

proefondervindelijk met de klanten samen worden uitgewerkt en vraagt dus om een andere werkwijze dan de traditionele verbetering van bestaande en daarmee bekende activiteiten. De onvoorspelbaarheid van het resultaat laat hierdoor per definitie niet meer toe dat *requirements* vooraf tot in detail betrouwbaar kunnen worden vastgesteld. Dit wordt geïllustreerd door de ELQ case in bijlage 1.

- Veel vertaalslagen.

Een andere factor die de mismatch bevordert, zijn de vele vertaalslagen die de gebruikelijke methodieken in de Software engineering hanteren. Requirements zijn veelal het vertrekpunt, eventueel aangevuld met een startarchitectuur die beperkingen oplegt aan de ontwerpen. Vervolgens worden de requirements vertaald in logische ontwerpen, die op hun beurt vertaald moeten worden naar technische ontwerpen. De technische ontwerpen worden vervolgens omgezet in code. Dit is op zichzelf al een complex proces. Dit probleem wordt slechts gedeeltelijk opgelost door de vertaalslagen in een serie van korte cycli (agile ontwikkelmethodes) te organiseren, zodat de aanhaking met de oorspronkelijke bedenkers en de uiteindelijke gebruikers niet volledig verloren gaat. Het probleem dat blijft is dat elke vertaalslag het risico op een 'vertaalfout' in zich draagt. Bovendien spreekt de opdrachtgever niet de taal van de logisch ontwerper, de logisch ontwerper niet die van de technisch ontwerper, en de technisch ontwerper zit niet altijd op dezelfde lijn als de programmeur. Daardoor kunnen ze elkaars producten vaak niet goed beoordelen en valideren, waardoor de foutkans alleen maar groter wordt. Hoe groter de afstand tussen de belevingswerelden van enerzijds de opdrachtgever en anderzijds de ontwerpers en programmeurs, des te groter is de kans op een flinke mismatch. Als dan ook nog de drijvende kracht achter dergelijke projecten de ICT organisatie is, wordt de betrokkenheid van de business geminimaliseerd, terwijl de software voor hen bedoeld is.

Verborgene bedrijfsregels

Een ander nadeel is dat *software engineering* er de gewoonte van heeft gemaakt om bedrijfsregels in de programmatuur te coderen zonder bijzondere maatregelen om ze snel te kunnen veranderen. Dit in tegenstelling tot gegevens, die in tabellen worden opgeslagen waardoor ze gemakkelijk te veranderen zijn. Bedrijfsregels worden gezien als de kern van de software die niet verandert. Dit belemmert de veranderbaarheid van software in hoge mate. In omgevingen waar een hoge mate van betrouwbaarheid is vereist en waar weinig verandert is dat misschien een goede strategie, maar in een sector waar marktbevingen telkens om andere producten, diensten en condities vragen is dat hinderlijk. Bedrijfsregels zijn als in beton gegoten. Iets in beton laten storten brengt veel rompslomp met zich mee, maar iets wat in beton is gestort laten veranderen is zo goed als onmogelijk. Gartner merkte in dit verband al diverse jaren geleden op: "*You cannot code your way into the future*".

Kortom, *software engineering* is een technologie gedreven paradigma, en dat past helemaal niet meer als niet de interne medewerker maar de klant of partner rechtstreeks aangesproken moet worden. Rechtstreeks met klant of partner communiceren vergt een hoge mate van automatisering van kennis (waaronder bedrijfsregels) en brengt een hoge mate van onzekerheid en veranderlijkheid met zich mee.

Noodzaak voor een ander paradigma

Digitalisering van de dienstverlening vraagt om een ander paradigma, met vier cruciale elementen:

1. de software moet kunnen communiceren met klanten en andere gebruikers, die niet van tevoren bekend zijn en geen opleiding in het gebruik van de software kunnen volgen; dit vraagt om een intelligente dialoog op basis van bedrijfsregels;
2. de software moet in staat zijn om routinematige beslissingen te nemen (dat deed voorheen de medewerker); dit vraagt om beslissingsintelligentie (bedrijfsregels);
3. de software moet eenvoudig en snel aan te passen zijn in nauwe samenhang met aanpassingen in de operationele organisatie; substantiële verbeteringen moeten binnen één tot vijf weken kunnen worden gerealiseerd, innovatiestappen mogen niet meer dan enkele

maanden en hooguit een half jaar kosten en bovendien moeten veel voorkomende optimalisaties (bijvoorbeeld herverdelen van werk) direct kunnen worden doorgevoerd;

4. de ontwikkelaanpak moet in hoge mate evolutionair zijn, zodat het resultaat van aanpassingen direct kan worden getest en gesimuleerd en bovendien snel in productie kan worden genomen.

Als de bedrijfspraktijk in hoge mate moet worden opgenomen in software dan wordt *engineering* van software feitelijk *engineering* van business. Niet de techniek maar de business moet dan leidend en sturend zijn. Het ontwerpen van de bedrijfspraktijk is de verantwoordelijkheid en competentie van op de bedrijfspraktijk georiënteerde mensen zoals organisatie- en materiedeskundigen, die een vergaande kennis hebben van en betrokkenheid hebben bij de bedrijfspraktijk.

Business Engineering

Business engineering is het antwoord op zowel de noodzaak voor een ander, niet door technologen gedreven paradigma, als op de noodzaak voor een gecombineerde discipline voor het beheren van bedrijfsregels en -processen.

Business engineering is een discipline die gericht is op de integrale en wendbare inrichting van de bedrijfspraktijk op een door de business gedreven manier.

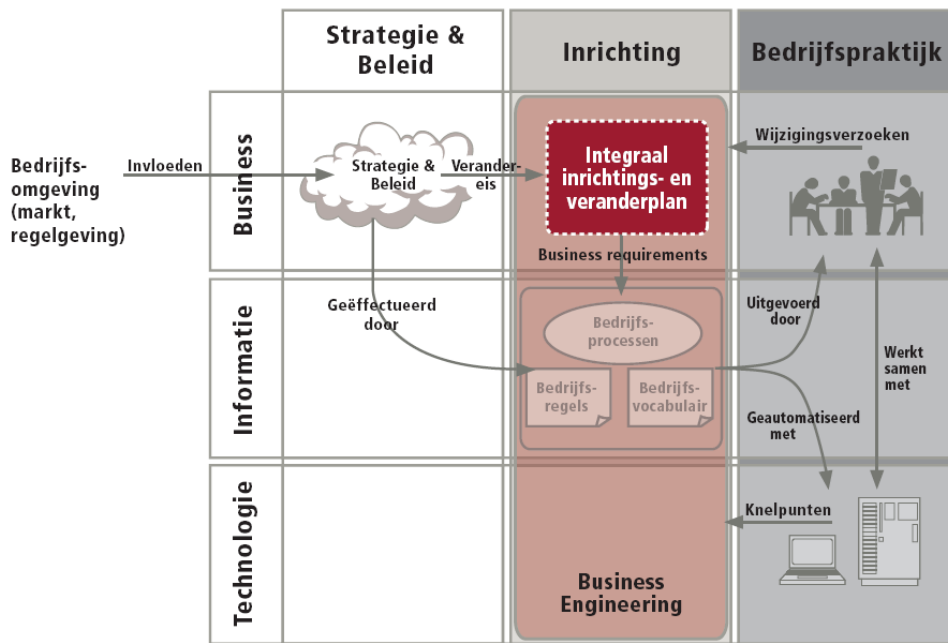
Kenmerkend voor deze discipline zijn:

- de integrale door de business gedreven veranderstrategie die technologie en mensenwerk als één geheel beschouwt;
- de techniek die erop gericht is om de conceptuele, niet-technische modellen direct door generieke *software engines* te laten uitvoeren zonder dat daar nog een technische vertaalslag voor nodig is;
- de hoge mate van automatisering van bedrijfskennis o.a. in de vorm van bedrijfsregels;
- de rol van de business deskundige (*business engineer*) die leidend is, terwijl de rol van de technologische deskundige op de achtergrond verdwijnt en alleen nog ondersteunend is;

Hieronder worden enkele van deze kenmerken nader toegelicht.

Integrale veranderstrategie

Business Engineering is erop gericht om veranderingen op een integrale manier te ontwerpen en realiseren. In Figuur 2 wordt dit geïllustreerd aan de hand van het informatiemanagement model uit hoofdstuk 1.



Figuur 2: **De Business Engineering aanpak**

Veranderingen die worden ingegeven door veranderende omstandigheden voor het bedrijf worden aan de hand van strategie en beleid ingezet met een integraal veranderplan dat zowel de business-, informatie- als technologische aspecten dekt. Onderdeel van dat plan zijn processen, rollen, bedrijfsregels en bedrijfsvocabulaire. Het voordeel van deze aanpak is dat de keuze tussen wat er wel en wat er niet geautomatiseerd wordt op een planmatige manier wordt gemaakt. Op die manier kan er gekozen worden voor vergaande automatisering van processen, en kunnen de overgebleven handmatige handelingen perfect worden afgestemd op de geautomatiseerde onderdelen. In plaats van dat software alleen als primair hulpmiddel fungeert gaan software en mensen samenwerken.

Directe uitvoering van modellen

Business en computer zouden dezelfde taal moeten spreken zodat de afhankelijkheid van meerdere technische tolken, met onvoldoende kennis van de inhoud, vervalft. Hierdoor wordt de business zelf in staat gesteld om de inrichting van de bedrijfspraktijk met inbegrip van de vergaande automatisering ervan te managen, met beduidend minder afhankelijkheid van de ICT. *Business engineering* is erop gericht om de specificaties van de te realiseren software te laten verwoorden in conceptuele, niet technische maar functionele modellen of specificatietalen. Business vocabulaire, processtromen, bedrijfsregels, taaklogica, communicatielogica, samenwerkingslogica en integratielogica (interactie met andere systemen via services) zijn elementen van deze modellen die de bedrijfskennis representeren. De modellen moeten begrijpelijk zijn voor mensen uit de bedrijfspraktijk en tegelijk formeel genoeg om door de computer te kunnen worden 'begrepen', ofwel de computer kan automatisch het door het model beschreven gedrag produceren².

Business engineering werkt het beste als gebruik wordt gemaakt van generieke *business engines* (zoals een *BPM Engine* of een *Rule Engine*), die in staat zijn de modellen van bedrijfskennis op dynamische wijze uit te voeren zonder verdere vertaalslagen. Het betreft uitermate complexe generieke software die op metamodel niveau opereert en die vanwege het generieke karakter tot de IT infrastructuur behoort. De prijs die hiervoor wordt betaald is dat er beduidend meer rekenkracht van de computer

² Men spreekt in dit verband ook wel van een vijfde generatietaal (5GL). Zie http://www.computable.nl/artikel/ict_topics/mobility/1855254/1277034/5gl-springlevend.html en http://www.computable.nl/artikel/ict_topics/development/1880473/1277180/5gl-springlevend.html.

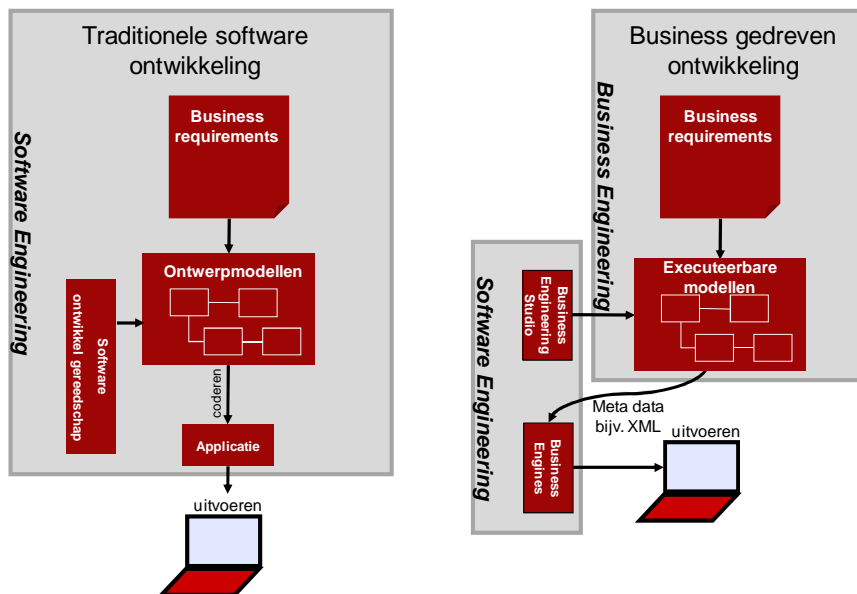
wordt gevraagd zodat complexe transacties die in hoge volumes moeten worden uitgevoerd een probleem kunnen vormen³.

Het grote, evidente voordeel van *business engineering* is dan ook dat de verandersnelheid erg groot kan zijn. Bedrijfsregels, processen of vocabulaires kan men aanpassen, direct het resultaat ervan beoordelen en zodra dat goed wordt bevonden direct laten uitvoeren. Dat bevordert een ontwikkelaanpak die veel meer evolutionair en in kleine stapjes kan plaatsvinden; omdat veranderingen snel zijn in te voeren, is het ook niet nodig om alle veranderingen grootschalig aan te pakken. *Business engineering* impliceert een evolutionaire ontwikkelmethodiek, die lijkt op gevestigde wendbare ontwikkelmethoden voor software zoals *Dynamic Systems Development Method (DSDM)* en *Lean Software Development (LSD)*.

Hernieuwde rol voor software engineering

Dat betekent overigens niet het einde van *Software engineering*, dat zijn merites behoudt in omgevingen die minder veranderlijk of meer technisch van aard zijn. Er is niets mis met *software engineering* op zich, er is alleen iets mis met *software engineering* in de context van snel veranderende business contexten.

Om *business engineering* te faciliteren zal er gespecialiseerde software nodig zijn die het mogelijk maakt om bedrijfsmodellen inclusief bedrijfsregels op te stellen en ook te laten uitvoeren; dergelijke software zou een *Business Engineering Studio* genoemd kunnen worden. De hierboven genoemde *business engines* zullen door *software engineers* gemaakt moeten worden. Figuur 3 geeft het verschil aan tussen de traditionele software ontwikkeling en de business gedreven aanpak. Bij de traditionele aanpak strekt *software engineering* (de zeggenschap van de ICT-afdeling) zich uit vanaf de *requirements*, inclusief de modellen, de ontwikkelgereedschappen tot en met het coderen (programmeren) en de applicatie zelf. Bij de business gedreven aanpak is het opstellen van de *requirements* en het maken van de modellen een zaak geworden van de business, terwijl het maken van de hulpmiddelen (gereedschap om te modelleren plus de *business engines*) nog tot het domein van de *software engineering* behoren.



Figuur 3 *De rolverschuiving van software engineering door de opkomst van business engineering*

³ Een alternatief is het geheel automatisch genereren van software code vanuit het functionele model. Een eis die daarbij gesteld moet blijven worden is dat het functionele model direct uitvoerbaar blijft, zodat er kan worden gesimuleerd en getest in termen van het model (functioneel debuggen) en niet alleen in termen van de technische software code (technisch debuggen). Mede daardoor verschilt de *business engineering* aanpak fundamenteel van *Model Driven Architecture (MDA)* voor softwareontwikkeling van OMG (zie www.omg.org/mda/).

Daarom is er eerder sprake van een verschuivende rol van *software engineering*. *Software engineering* is niet goed in staat de wendbaarheid van dienstverlenende business te ondersteunen, maar des te beter in staat om de stabiele generieke software (modelleergereedschap en engines) te maken die nodig is om *business engineering* te ondersteunen.

De *software engineer* blijft ook nodig bij systeemontwikkeling, maar krijgt een ondersteunende rol, bijvoorbeeld voor het coderen van uitzonderlijke businesslogica en voor het aansluiten op andere systemen (ook wel aangeduid als *plumbing*). De specificaties hiervoor komen uit het model dat is gemaakt door een *business engineer*.

Impact van *Business Engineering* op het inrichten van de bedrijfspraktijk

De nieuwe rol van Business Engineer

De discipline van *business engineering* vraagt om *business engineers*. *Business engineers* zijn medewerkers met ervaring in en vergaande kennis van de bedrijfspraktijk van de betreffende sector, die daarnaast de nodige modelleercapaciteiten hebben. Zij verstaan de kunst van het abstraheren, formeel modelleren en specificeren naast de kunst van het communiceren met de business. Zij zijn daarbij zowel gericht op de organisatorische aspecten als op de daarbij aansluitende informatiesystemen. Bovendien kunnen zij zich snel inwerken in de essentie van een nieuw business domein.

Er wordt ook wel gesproken over ICT'ers met veel kennis van de business, maar het hebben van veel ervaring met *software engineering* kan juist een risicofactor zijn. De reden daarvoor is dat software engineers van nature uitgaan van een hoge mate van maakbaarheid van te ontwikkelen systemen: zij geloven in de effectiviteit van een blauwdrukaanpak. *Business engineering* vraagt om een aantal fundamenteel andere vaardigheden die technisch georiënteerde ICT'ers meestal niet in huis hebben, zoals sterke communicatieve vaardigheden, omgaan met onzekerheid en diepgaande kennis van een specifieke business.

Verschuiving van werk van ICT naar business

Het is te verwachten dat er een flinke verschuiving gaat optreden van *software engineering* naar het hierboven gedefinieerde *business engineering*. Dit betekent dat de business nieuwe verantwoordelijkheden op zich zal moeten nemen en dat ICT bestaande activiteiten zal moeten loslaten.

De business zal *business engineers* moeten aannemen en budget, dat voorheen aan ICT werd gespendeerd, moeten besteden aan het opzetten en onderhouden van eigen middelen die hiervoor nodig zijn. In de praktijk blijkt deze verschuiving niet helemaal vanzelf te gaan. Er ontstaan grote weerstanden van de kant van de ICT, en de business blijft soms gemakkelijk in de afwachtende rol. Wil men echter de wendbaarheid faciliteren die de business voor ogen staat, dan zal de verschuiving noodzakelijk worden.

De competenties die nodig zijn voor business engineering zullen veel meer dan nu een combinatie moeten zijn van technologische kennis, visionaire capaciteiten, modelleervaardigheden en kennis van de business⁴. De verwachting is dat slechts één op de drie huidige ICT'ers in staat zal zijn deze nieuwe rol te vervullen. De overigen zullen veel meer dan nu in de 'harde' software industrie hun rol moeten vinden, waar het *software engineering* paradigma blijft bestaan.

Overige organisatorische veranderingen

Om de kennis op het gebied van *BRM* en *BPM* op organisatieniveau te borgen zal er een horizontaal 'Business engineering expertisecentrum' moeten komen, dat verantwoordelijk is

⁴ Zie Niel MacDonald, "Programmers vs Machines", Gartner Symposium ITxpo 2008.

voor de standaarden wat betreft methoden, technieken, hulpmiddelen en kwaliteit op het gebied van beide disciplines. Dit laatste is een fundamenteel vereiste voor volwassenheidsniveau 3, zoals beschreven in hoofdstuk 4. Ook is dit expertisecentrum verantwoordelijk voor het beheer van overkoepelende procesraamwerken en bedrijfsregels. De scope en taken van een dergelijk expertisecentrum raken aan die van de discipline die *Enterprise Architectuur* (EA) wordt genoemd. In feite komen *BRM*, *BPM* en EA bij elkaar in *business engineering*. Het is niet onwaarschijnlijk dat de bestaande op applicatieontwikkeling gerichte onderdelen van IT organisaties geleidelijk zullen transformeren in een '*Business engineering* expertisecentrum'. Zoals hierboven reeds aangegeven zal de bemensing daarbij flink moeten veranderen⁵.

Aan elk *end-to-end* bedrijfsproces moet een 'proceseigenaar' met mandaat worden toegekend. Hierbij dient in aanmerking te worden genomen dat alleen grote bedrijven meerdere *end-to-end* primaire bedrijfsprocessen hebben. De werkeenheden, die onderliggende werkprocessen uitvoeren, worden dienstverleners die SLA's afsluiten met de proceseigenaren en daarop ook worden afgerekend. Ook wat dit betreft is er sprake van een gevoelige machtsverschuiving.

Per *end-to-end* bedrijfsproces is er behoefte aan een 'bedrijfsproces ontwikkel- en verbeterteam' (BPT), geleid door een 'procesoperatie manager' die in opdracht van de proceseigenaar voortdurend bezig is met verbetering en verandering van de bedrijfspraktijk. Als er, zoals bij minder grote bedrijven, sprake is van slechts één primair *end-to-end* bedrijfsproces, komen BPT en 'Business Engineering expertise centrum' met elkaar overeen. De veranderingen die het BPT uitvoert zijn afkomstig van knelpunten die het operationele management niet direct zelf kan oplossen en van veranderingen in strategie en beleid. Een BPT is een multidisciplinair team bestaande uit *business engineers*, business specialisten, *software engineers* en *enterprise* architecten die in staat zijn op businessniveau verschillende een *end-to-end* bedrijfsproces en daarbij behorende bedrijfsregels en bedrijfsvocabulaires te modelleren, specificeren en realiseren. Het BPT is uiteraard ook verantwoordelijk voor het verdiepen van de kennis in de informatiesystemen op basis van terugkoppeling uit de bedrijfspraktijk.

Men kan zich afvragen wat er in dit verband zal gebeuren met de rol van *CIO* en *COO*. Het ligt voor de hand om te veronderstellen dat deze rollen samen zullen vloeien. De operatie wordt immers grotendeels geautomatiseerd, waardoor deze verantwoordelijkheden niet goed meer zijn te scheiden. De *COO* zou de leiding van het centrale '*Business engineering* expertisecentrum' op zich kunnen nemen, maar daarvoor zou ook een aparte *Chief Change Officer* (*CCO*) kunnen worden benoemd⁶. Dit expertisecentrum is immers overkoepelend verantwoordelijk voor de kwaliteit van de inrichting van de bedrijfspraktijk. Uiteraard blijft er sprake van een *CTO* die verantwoordelijk is voor alle infrastructurele ICT aspecten.

Wat nu?

De hierboven geschetste vergaande organisatorische veranderingen bij het inrichten van de bedrijfspraktijk kunnen zelfs voor innovatief ingestelde mensen een schrikbeeld zijn. De volgende reactie ligt voor de hand: het klinkt mooi, wie zou dit eigenlijk niet willen, maar dat krijgen we nooit voor elkaar in onze ambtelijk ingestelde organisatie, of in deze turbulente marktomstandigheden.

Allereerst moet in gedachten worden teruggeroepen dat wat hier beschreven is gaat over volwassenheidsniveau 3, een niveau dat voor veruit de meeste organisaties nog (verre) toekomstmuziek is. Vanwege de vergaande veranderingen moet er een evolutionair groeipad worden gevolgd. Daartoe kunnen twee zeer uiteenlopende wegen worden gevolgd, *green field* of *metamorfose*, die beiden een actieve betrokkenheid van de top van het bedrijf vergen.

- *Green field*;
Richt een zelfstandig opererende nieuwe business unit op die stapsgewijs maar snel op

⁵ Deze beweging wordt ook door Gartner gesignaleerd; zie de samenvatting daarvan in Mahoney, John & Gomolski, Barbara, "The Changing Shape of IT: What We've Uncovered, Where You Can Find It" (27 mei 2008)

⁶ <http://tlir.wordpress.com/2007/05/22/the-case-for-the-chief-change-officer/>

volwassenheidsniveau 2-3 wordt gebracht. Hierbij worden partners gezocht met een bewezen state-of-the-art kennis op het gebied van geïntegreerde toepassing van *BPM* en *BRM* in de markt. Het is mogelijk dat deze partners een permanente ondersteunende rol blijven spelen.

Dit is wat Lehman Brothers met ELQ heeft gedaan voor een nieuwe markt (zie ELQ case in bijlage 1). Als de nieuwe business unit echter op dezelfde markt moet gaan opereren als de traditionele business, moet worden toegelaten dat er echt wordt geconcurrereerd (dit vergt veel lef van het topmanagement). Als de nieuwe business veel succesvoller blijkt te zijn, loopt de traditionele business vanzelf terug en kan uiteindelijk worden opgeheven. Het geschikte personeel kan geleidelijk overstappen naar de nieuwe business unit (en wil dit dan ook graag), maar er zal ook het nodige op routinewerk ingesteld personeel en personeel dat zich moeilijk laat omscholen, moeten afvloeien.

Een voorbeeld van een dergelijke ontwikkeling vormt Ditzo van Fortis⁷ ("We hebben Ditzo van de grond af opgebouwd met nieuwe mensen, systemen en samenwerkingspartners. We hebben geen last van vastgeroeste structuren en overtollige managementlagen en kunnen daardoor vrijuit ondernemen tegen lage kosten met de steun van onze sterke aandeelhouder Fortis").

- *Metamorfose*;

1. Kies een geschikt operationeel werkproces om te migreren naar volwassenheidsniveau 2. Geschikte werkprocessen zijn processen die betrekking hebben op distributie of levering van individuele massaproducten en -diensten met voldoende complexiteit, zoals bijvoorbeeld verkopen van hypotheeklen of aanbieden van individuele WMO⁸ producten. De inrichting van het huidige werkproces bevindt zich bij voorkeur al op volwassenheidsniveau 1. 'Geschikt' houdt ook in dat er veel te verbeteren valt met als resultaat daadwerkelijk toegevoegde waarde voor het bedrijf. Dit moet weloverwogen worden vastgesteld vanuit bedrijfsstrategie en -beleid en veel minder zoals gebruikelijk door het beleid van ICT. Veelal vergt dit de ontwikkeling van een nieuwe informatiesysteem met als initiële scope het gekozen werkproces.
2. De betreffende business unit (en dus niet de ICT afdeling) gaat vervolgens *business engineers* aannemen of opleiden of gaat een partnership aan met een dienstverlener op de markt die heeft bewezen met succes *BPM* en *BRM* integraal te kunnen introduceren, minimaal op volwassenheidsniveau 2. Deze mensen realiseren projectmatig, in nauwe samenwerking met de business unit en met de ICT afdeling, de ontwikkeling en implementatie van het nieuwe informatiesysteem en de organisatorische overgang naar volwassenheidsniveau 2.
Het topmanagement moet steeds bereid zijn om doelgericht in te grijpen bij de te verwachten onenigheid tussen business en dienstverlener enerzijds en de gevestigde ICT anderzijds. Daarbij dienen de evidente belangen van de business het zwaarst te wegen. Het volgen van een *middle-out* aanpak zoals beschreven in de laatste paragraaf van hoofdstuk 4 ligt hierbij voor de hand, omdat er dan op effectieve wijze prioriteiten kunnen worden gesteld.
3. De opgebouwde kennis wordt vervolgens geborgd door de oprichting en bemensing van een permanent 'bedrijfsproces ontwikkel- en verbeterteam' (BPT) gericht op het *end-to-end* bedrijfsproces waarin het oorspronkelijke werkproces een belangrijk element vormt. Ook worden een proceseigenaar (met voldoende mandaat m.b.t. de betrokken business units) en procesoperatie manager (geeft leiding aan het BPT) benoemd. Het BPT realiseert in nauwe samenwerking met de betrokken business units, op evolutionaire wijze de verbreding van het informatiesysteem en de organisatorische overgang naar volwassenheidsniveau 3. De opgebouwde kennis van het BPT wordt daarbij geborgd door de oprichting van een '*Business engineering* expertisecentrum', dat deels wordt bemenst met leden van het BPT. Het BPT blijft natuurlijk wel bestaan want het aanvankelijke bedrijfsproces moet verbeterd blijven worden. Er wordt een

⁷ www.ditzo.nl

⁸ WMO = Wet Maatschappelijke Ondersteuning, die betrekking heeft op thuiszorg en hulpmiddelen voor ouderen en gehandicapten.

Chief Change Officer benoemd die de leiding van het expertisecentrum op zich neemt. Hierbij dient in aanmerking te worden genomen dat alleen grote bedrijven meerdere primaire *end-to-end* bedrijfsprocessen hebben. Als dit niet het geval is komen BPT en '*Business engineering* expertisecentrum' met elkaar overeen.

4. Ondersteund door het '*Business engineering* expertise centrum' wordt stapsgewijs een ander *end-to-end* bedrijfsproces onderhanden genomen.

Een goed voorbeeld van een succesvolle metamorfose vormt de LEI case uit bijlage 1. Omdat *BRM* destijds nog in de kinderschoenen stond en omdat *BPM* nog niet was uitgevonden, heeft LEI alle benodigde kennis zelf opgebouwd, inclusief de benodigde *BRM* en *BPM* technologie.

Het lijkt erop dat de 'green field' route bij grote ambtelijke organisaties sneller, goedkoper en zekerder tot resultaat leidt dan de metamorfose. Metamorfose van dergelijke versterde organisaties komt vaak pas onder grote druk te laat op gang en kost dan teveel tijd en geld. Als de druk van de markt en/of wet- en regelgeving erg groot wordt is er geen andere keus meer dan een echte frisse wind te laten waaien (green field). Het bloed kruipt immers toch waar het niet gaan kan.

Uit: Alcedo Coenen, Leo Hermans, Matthijs van Roosmalen en Silvie Spreeuwenberg, "Uw bedrijf geregeld met Business Rule Management". Uitgegeven door SDU, Den Haag. Het boek is deel 38 van de serie "ICT Bibliotheek" onder het label Academic Service. ISBN 9789012125956. Met toestemming van de uitgever gereproduceerd in Via Nova Architectura.