

De solution development cycle

Van IT-afdeling naar 'professional solution organisation'



Hugo Meijers

Managers stellen onder invloed van het buzzwoord 'solutions' steeds hogere eisen aan de kwaliteit van dienstverlening van hun interne IT-afdelingen. Die hebben echter grote moeite om de snelheid van veranderingen in bedrijfseisen en technische innovaties bij te benen. Hugo Meijers beschrijft in dit artikel een nieuwe manier van denken over hoe IT-afdelingen zich kunnen organiseren voor innovaties.

'Solutions' of oplossingen zijn de nieuwe mantra's voor het verkopen van combinaties van bestaande en nieuwe technologieën. Teneinde toevoegende waarde te kunnen blijven bieden, en zodoende bestaansrecht te behouden, moeten IT-afdelingen volwassener worden in het adopteren en in bedrijfscontext plaatsen van nieuwe technologie. Het capability maturity model (CMM) vervult hierin een schijnbare rol en is gedefinieerd als de implementatie van standaard- en geformaliseerde processen. Deze processen dienen garant te staan voor kwaliteit en snelheid van dienstverlening, zowel qua ontwikkeling, levering als onderhoud van technologie. De principiële aanname in CMM is dat productkwaliteit voortvloeit uit procesbeheersing, echter geen enkel bewijs voor deze causaliteit bestaat. Standaardisatie en formalisering maken bedrijven rigide en bureaucratisch, daar waar flexibiliteit en aanpassingsvermogen geboden is.

Procesbeheersing

CMM is diep geworteld in het total quality management-concept waarin centraal staat dat productkwaliteit (mede) bepaald wordt door de kwaliteit van de procesbeheersing. Hierin is 'proces capability' gedefinieerd als het statistisch begrippen van de verwachte prestatie. CMM-

ontwerpprincipes zijn herhaalbaarheid, meetbaarheid, beheersing en standaardisatie (voorspelbaarheid). Deze principes zijn afkomstig uit massaproductie en bureaucratische modellen. Tot op heden bestaat er geen feitelijke bewijs dat deze industriële principes daadwerkelijk zorgdragen voor een betere productkwaliteit.¹

In een recent gehouden workshop² met CMM-niveau 4 en 5 IT-afdelingen is een belangrijk punt naar voren gebracht. Het betrof de onzekerheid, vaagheid en onduidelijkheid over de betekenis van 'maturity'. Binnen CMM wordt maturity uitgelegd als de mate waarin processen standaard, geformaliseerd en zorgvuldig vastgelegd zijn. Deze enge definitie van maturity heeft bijgedragen aan een immer nog beperkt beeld van wat verstaan kan worden onder kwaliteit van dienstverlening. Volwassenheid van een IT-afdeling hangt samen met de mate van waarde die zij bijdraagt aan de bedrijfsvoering, onafhankelijk op welke wijze dit totstandkomt.

Onderdeel van de begin jaren negentig heersende ISO 900x-hype is het auditten van bedrijfs-systemen door derde partijen. De simpelste manier om procesbeheersing vast te stellen is via de verificatie of vastgelegde beschrijvingen daadwerkelijk worden gevolgd. Er wordt nauwelijks gekeken naar de daadwerkelijke product-



kwaliteit. Deze onafhankelijke evaluatie heeft bijgedragen aan de interpretatiewijze en implementatievorm van kwaliteitsnormen. Het heeft invloed gehad op de mate en detail van vastlegging van processen. Deze interpretatie is contra de huidige managementdenkwijze. Die wordt gedomineerd door het afbreken van bureaucratie en het afstand nemen van standaardisering en formalisering. Aan de ene kant heerst chaos, aan de andere rigiditeit. Het overgangsgebied wordt veelal aangeduid met de 'edge of chaos'.

de principiële aanname in CMM is dat productkwaliteit voortvloeit uit procesbeheersing, maar voor deze causaliteit bestaat geen enkel bewijs

Volgens Russell Ackoff³ is het meestal slechter om het onjuiste op een juiste wijze uit te voeren dan het juiste op een onjuiste wijze. In de systeemleer is meestal het object belangrijker dan het proces in het bereiken van een kwalitatief resultaat. Dit is uiteraard geheel afhankelijk van de definitie van het containerbegrip 'kwaliteit'. In de context van oplossingen is de heldere afbakening en begripsvorming van het object belangrijker dan het proces van totstandkoming.

Resteert de vraag over de zinvolheid van vastlegging van standaardprocessen en het toekennen van vaste rollen en verantwoordelijkheden. Dit is slechts zinvol bij zich herhalende processen in een stabiele omgeving. In werkelijkheid hebben, in een tijd van massamaatwerk en een continu veranderende omgeving, slechts weinig processen deze eigenschap.

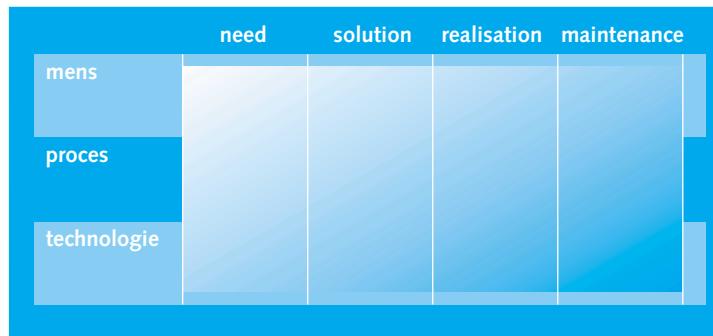
Werkbenadering

De wetenschap rond complexiteit heeft nieuw inzicht verschaft in de voorspelbaarheid van processen. Het helpt tevens bij het begrijpen van de mate van herhaling van processen (bijvoorbeeld door toepassing van fractals). Om complexiteit beter te doorgronden moet een onderscheid gemaakt worden tussen soorten processen, en tussen proces en inrichting. Er zijn twee soorten processen: kennisprocessen en fysieke processen. Processen binnen IT-afdelingen zijn kennisprocessen. Bij kennisprocessen gaat het niet zozeer om standaardisatie of inrichting. Kenniswerk is van nature een conceptuele activiteit van creativiteit en verbeeldingskracht. Hierin zijn context en relevantie belangrijker dan aanpak en middel. De focus dient te liggen op wat er moet gebeuren, en niet hoe. Standaardisatie van processen generaliseert context en gaat voorbij aan relevantie. Het versimpelt de complexiteit van de werkelijkheid tot een irreëel en daardoor onpraktisch deel van die werkelijkheid. De werkelijkheid wordt gereduceerd tot enkelvoudig zwart-wit.

Systeem- en cybernetische theorieën geven aan dat organisaties zijn opgebouwd uit drie hoofdelementen: structuur, processen en patronen. Tot dusver wordt organisatiestructuur beschouwd als een clustering van functies en als iets statisch (in evenwicht), wat resulteert in hiërarchische machtsbouwwerken met inflexibele en standaardprocessen. Deze beschouwingwijze is ver van de realiteit. Verkeerde aandacht wordt gegeven aan structuur en proces, en weinig tot geen aan patronen. Technologieën als intranet en e-mail veroorzaken informele en verschuivende patronen van samenwerking. Deze informele patronen bepalen steeds vaker wat wel en wat

standaard	gestructureerd
uniform	maatwerk
naleving centraal	relevantie centraal
contextalgemeen	contextspecifiek
controle van gedrag	controle van resultaat
lineair, sequentieel	cyclisch, iteratief
onafhankelijke actoren	samenwerkende actoren

Tabel 1: Belangrijkste verschillen tussen standaard- en gestructureerde processen



Figuur 1: De solution development cycle

niet gedaan wordt, alsmede de wijze waarop. Bij deze patronen hoort een denken in structuur en niet in standaard (zie tabel 1).

Kenniswerk moet niet gestandaardiseerd worden, maar gestructureerd. Net als bij sport dienen de grenzen en spelregels aangegeven te zijn, maar het spel mag niet gehinderd worden. Buiten het spel zijn managers aangewezen om professionals te trainen, tijdens het spel zijn ze er hooguit om te begeleiden. Dan is het aan de professional om er iets van te maken. Niet de formele, maar informele patronen binnen de organisatie bepalen dan snelheid en kwaliteit van handelen en resultaat.

De SDC

Een 'solution' is de bevrediging van een behoefte in termen van *mens*, *proces* en *technologie*. In deze context is de mens (van directeur tot de werkvloer) de gebruiker. Met proces wordt het bedrijfsproces bedoeld dat ondersteund wordt door technologie. De technologielaag behelst alle vormen van technische inrichting, van telefoon tot geavanceerde werkstations. Deze drie lagen vormen een *integraal geheel*, een interafhankelijk systeem. Iedere laag is zelf ook weer opgebouwd uit drie lagen (mens, proces en technologie).

Binnen de technologielaag bijvoorbeeld is de mens de IT-professional, het proces de systeemontwikkeling en de inrichting de ontwikkelomgeving.⁴ Het operationele werk, ofwel het primaire proces binnen ontwikkelomgevingen zoals die van IT- en R&D-afdelingen, is weergegeven in de solution development cycle (SDC) (zie figuur 1). De SDC bestaat uit vier hoofdfasen: *need*, *solution*, *realisation* en *maintenance*.

De *need-fase* kan worden beschouwd in het verleden, het huidige en de toekomst. Voor het verleden en het huidige zijn begrippen als 'probleem' of 'vraagstuk' gebruikelijk. Voor aspecten die met een gewenste situatie te maken hebben gebruikt men vaak de termen 'kans', 'behoefte' of 'wens'.

In alle gevallen gaat het er in deze fase om een bewustzijn te creëren en een eerste inzicht te verschaffen in de waarde van een ICT-gerelateerde oplossing, zonder daadwerkelijk op een mogelijke oplossing zelf in te gaan. Probleemanalyse, 'visioning' en strategisch plannen behoren tot deze fase.

De *solution-fase* gaat over het definiëren van een antwoord op de in de need-fase bepaalde vraag. Dit antwoord wordt zowel kwalitatief als kwantitatief in algemene termen verwoord. Voor een IT-afdeling gaat het om de technische component van een oplossing, alhoewel rekening gehouden dient te worden met een oplossing zonder technologisch component. Afhankelijk van de complexiteit van de vraag kunnen meerdere scenario's, bijvoorbeeld van behoudend tot agressief, worden beschreven. De beschrijving van de oplossing is vrij van jargon en in bewoording van de gebruiker c.q. de vraageigenaar. De beschrijving dient voldoende detail te bevatten om zinvolheid, haalbaarheid en wenselijkheid van de oplossing te beoordelen. De mate van detail om daadwerkelijk systemen te bouwen dan wel in te richten, wordt pas in de volgende fase bereikt.

Onderdeel van deze fase is een implementatiestrategie met daarin op hoofdlijnen de aanpak, organisatie, benodigde vaardigheden en tijdslijnen. Het resultaat van de solution-fase is een *waardepropositie* met daarin de voordelen en risico's in zowel kwalitatieve als kwantitatieve termen.

De *realisation-fase* behelst het verder in detail definiëren en ontwikkelen van de oplossing, het testen, accepteren en implementeren. Deze fase komt overeen met de standaard maatwerk-ontwikkelmethoden dan wel de applicatie-implementatiemethoden. Deze fase eindigt met het in productie nemen van de oplossing in haar totaliteit.

	need-fase	maintenance-fase
meervoudige:enkelvoudige skills	80:20	20:80
cycli per solution	in weken/maanden	in uren/dagen
aantal projecten	weinig	veel
oriëntatie	resultaat	inspanning
focus	effectiviteit	efficiëntie
interafhankelijkheden	veel	weinig
planning	organisch (los)	strak en beheersmatig

Tabel 2: Glijdende schaal van need naar maintenance

De laatste fase is het operationeel beheren en op niveau houden van de oplossing (*maintenance*). Hieronder vallen ITIL-processen als probleem-, change-, incident- en helpdeskmanagement.

Een significante verandering (in termen van kosten of impact) dient te allen tijde via de volledige SDC te lopen. Hetzelfde gedachtegoed, uiteraard in de juiste context geplaatst van de aard van de verandering, kan worden toegepast. Het probleem, de gewenste oplossing en een beheerste ontwikkeling helpen ad hoc systeemaanpassingen te voorkomen.

De verschillen in type werk en aandachtsgebieden tussen de vier fasen is opgenomen in tabel 2. Belangrijk in de SDC is de overgang van de solution- naar de realisation-fase. Het komt nog steeds veel voor dat de gebruikersorganisatie, of externe management- en businessconsultants, voorfasen uitvoeren zonder betrokkenheid van professionals die belast zijn met de realisation-fase. Deze ont koppeling resulteert in een afwijkende interpretatie van wat bedoeld is in de waardepropositie. Het niveau van abstractie in de waardepropositie is van dien aard dat het niet direct als basis voor systeembouw kan dienen.

De cyclus kan een indruk van lineair en sequentieel handelen wekken. Dit is niet de bedoeling. Gedurende alle fasen dient continu teruggegrepen te worden naar eerdere beslissingen en afwegingen. Zeker in de eerste fasen is dit cruciaal voor het creatieproces. Dit betekent niet dat er geen voortgang geboekt wordt door cirkelredentatie. Voortschrijdend inzicht wordt geborgd door drempelwaarden in het proces in te bouwen. Indien een drempelwaarde overschreden wordt, kan de volgende fase of stap worden aan-

gevangen, meestal parallel met activiteiten in de eerdere fasen. Tevens is het van belang om de sleutelrollen gedurende de volledige tot stand wording van de oplossing betrokken te houden. Behoud van kennis en inzicht in het doorlopen proces voorkomt heroverwegingen en misinterpretatie omtrent eerder genomen beslissingen.

Succesvolle ontwikkeling

Wat bepaalt nu het succes van het ontwikkelen van oplossingen? Organisaties en bedrijven zijn complexe systemen waarbij een integrale systeembenadering bepalend is voor het oplossen van de vraag. Door de samenhang van mens, proces en technologie betekent dit een projectmatige en multidisciplinaire werkwijze. De effectiviteit van deze aanpak wordt hoofdzakelijk bepaald door een gemeenschappelijk referentiekader. Dit houdt een gedeelde vocabulaire en aanpak in, zodat werkwijze, beschrijvingswijze en denkwijze op elkaar zijn afgestemd. In de praktijk ontbreekt dit.

Een complicerende factor is het gebruik van externe consultants die kennisdiscontinuïteit veroorzaken. Dit staat los van de kloof tussen de solution- en realisation-fasen. Deze kloof wordt nog eens nader beschreven in tabel 3. De eerste twee fasen van de SDC hebben hoofdzakelijk een businessoriëntatie, de laatste twee een IT-oriëntatie.

Zoals aangegeven kan de SDC worden toegepast op twee categorieën oplossingen; oplossingen die te maken hebben met huidige systemen en oplossingen die te maken hebben met nog niet bestaande systemen. De aanpak binnen de SDC is fundamenteel anders voor deze twee categorieën. Dit alleen al maakt standaardisatie zinloos. Nieuwe oplossingen, afhankelijk van hun com-



Noten

1. Sommige fervente voorstanders van CMM wijten dit aan de interpretatiewijze van CMM. De praktijk spreekt voor zichzelf, de rest kan verbannen worden naar meningschap.
2. Mark Paulk and Mary Beth Chrissis, November 1999 High Maturity Workshop, March 2000, Special report CMU/SEI-2000-SR-003.
3. Russell Ackoff: *Re-creating the corporation. A design of organizations for the 21ste century*, Oxford University Press, 1999.
4. Een nieuw fenomeen, de professional services automation (PSA)-platformen zijn een voorbeeld van inrichting. Deze platformen kunnen gezien worden als een ERP-systeem voor project-omgevingen binnen de zakelijke dienstverlenings-sector.

plexiteit, moeten ontworpen worden. Innovatie, risico nemen, experimenteren en improviseren behoren hierbij. Analyse heeft geen zin, anders dan het inschatten van de impact op het bestaande. Vaardigheden in synthese en inductie zijn belangrijk. Bij bestaande oplossingen kan meer mechanisch te werk worden gegaan. Afbakening en sequentieel ontwikkelen zijn begrippen die bij deze categorie horen.

Hoe moet je nu organiseren voor innovatie? Inbedden in een hiërarchie van functies zal de SDC alleen maar reduceren tot een bureau-cratisch systeem met veel overdrachtsmomenten. Het inbedden in de algemeen voorkomende hiërarchische structuur van IT-afdelingen zal daarom niet werken, zeker niet voor de eerste twee fasen van de SDC. Hiervoor is een open systeembenadering nodig van zelfsturende, multidisciplinaire en gelijkwaardige teams. Teamsamenstelling bestaat uit relatie-, architectonische en programmamanagementrollen die worden ingevuld door gebruikers en IT-professionals. Verantwoordelijkheden kunnen per fase verschillen. De wisseling van sleutelfiguren moet tot een minimum beperkt blijven. Het is zelfs mogelijk een pool van professionals samen te stellen zonder functionele afbakening. Binnen deze pool worden opdrachten op basis van inschrijving aan individuen toegewezen. Dat gebeurt op basis van motivatie, expertise en competentie.

De solution lifecycle

Tot slot nog een opmerking over de relatie met de solution lifecycle. De solution development cycle is niet hetzelfde als de solution lifecycle. Vaak worden deze begrippen ten onrechte door elkaar gebruikt. De lifecycle is de absolute (meestal aangeduid met versienummer) dan wel relatieve (ten opzichte van het bedrijf) fase waarin een oplossing verkeert. Het gaat om een oplossing in productie en komt overeen met de maintenance-fase. Echter, de maintenance-fase zegt niets over de leeftijd c.q. nieuwigheid van de oplossing.

Vanuit het oogpunt van beheersing en waarde-toevoeging dienen oplossingen afgeschreven te worden, zowel technisch als economisch. Slechts weinig bedrijven hebben hier een systematiek voor. Het actief en consistent deactiveren van informatiesystemen is nog onontgonnen gebied. Het proces van informatieplanning geeft hier invulling aan. In dit proces worden technisch en economische afwegingen gemaakt voor het aanpassen, upgraden dan wel deactiveren van informatiesystemen. De resultaten van deze afwegingen dienen als input voor de solution development cycle.

	businessoriëntatie	IT-oriëntatie
scope (omvang)	groot, bedrijfssystemen	beperkt, vaak één afdeling
diepgang	globaal, hoofdlijnen	groot, veel detail
aanpak	top-down, vanuit totaal	bottom-up
taal	businessjargon	IT-jargon
afbeeldingwijze	flow, stappen per afdeling	ERD en DFD
insteek	bewustzijn, begrip en inzicht	completeheid
doelstelling	totstandwording product	steeembouw
transformatiefactor	product van bedrijfssysteem	informatiestromen en verzamelingen

Tabel 3: Verschillen tussen business- en IT-oriëntatie