



De sleutel tot het beheersen van complexiteit

White paper

Hoe besluitvorming in ICT de complexiteit van het applicatielandschap proactief kan beheersen

Inhoudsopgave

| | |
|----------------------------|----|
| Samenvatting..... | 3 |
| Introductie..... | 4 |
| Geschiedenis | 4 |
| Wat is complexiteit? | 5 |
| De problemen..... | 6 |
| Succesfactoren | 7 |
| Conclusie | 10 |
| De auteur..... | 10 |
| Feedback | 10 |

Samenvatting

Complexiteitsreductie in applicatielandschappen wordt doorgaans ingestoken vanuit de techniek, waarbij de onderliggende oorzaken worden genegeerd. Dat is echter geen duurzame aanpak gebleken, complexiteit is immers mensenwerk: de complexiteit neemt met het verstrijken van de tijd weer toe. Maar daar is wel een remedie tegen.

Deze white paper geeft ICT-managers, Businessopdrachtgevers van ICT-projecten, applicatie-eigenaars, programma- en project managers, architecten en anderen die zijn betrokken bij applicatielandschappen inzicht in het begrip 'complexiteit', de problemen met complexiteit en de succesfactoren voor een blijvende oplossing van het complexiteitsvraagstuk.

Complexiteitsreductie is slechts blijvend effectief als tegelijkertijd aandacht wordt besteed aan de beheersing van verdere complexiteitstoename. In die beheersing speelt ICT-Governance, de besluitvorming rondom ICT, een grote rol.

Als ICT-Governance wordt beschouwd vanuit het oogpunt van een sociaal systeem, *mensen* die met elkaar *interacteren*, ontstaat een diepgaand inzicht in de oorzaak van complexiteitstoename. En het belangrijkste, waar de aanknopingspunten zitten voor een blijvende oplossing.

De vragen die deze paper beantwoordt:

- Welke problemen veroorzaakt complexiteit in het applicatielandschap?
- Welke succesfactoren zijn van belang om complexiteit van het applicatielandschap structureel te beheersen?

Introductie

Door fusies, overnames, innovaties, wet- en regelgeving en veelbelovende (ERP-)systemen zijn applicatielandschappen uitgegroeid tot enorme proporties. Het landschap bestaat uit een mix van nieuwe applicaties en legacy-applicaties. De flexibiliteit is klein en de complexiteit is zó groot dat het landschap zélf de vooruitgang blokkeert. Beheer- en onderhoudskosten nemen toe, kleine aanpassingen kosten onverwacht veel geld. En het overzicht in het landschap neemt af: er is niemand meer die het gehele speelveld overziet.

De afgelopen jaren hebben veel programma's het licht gezien die zich bezig hielden met complexiteitsreductie. Deze zijn tot op heden niet succesvol gebleken. Dat komt doordat deze programma's veelal het symptoom - complexiteit - bestrijden door alleen het bestaande applicatielandschap onder de loep te nemen. Op zichzelf een logische aanpak, mits tegelijkertijd wordt voorkomen dat de complexiteit opnieuw toeneemt. En dat laatste ontbreekt doorgaans omdat organisaties in essentie niet weten hoe die complexiteit tot stand is gekomen. Men kent de root cause niet. In plaats daarvan wordt complexiteit als een gegeven beschouwd: 'onze organisatie is nu eenmaal complex, we hebben veel afdelingen, lastige toezichthouders, grote programma's en veel applicaties'.

Complexiteit is echter geen gegeven. Er is wel degelijk een root cause aan te wijzen, die door systematisch onderzoek kan worden geduid en vervolgens geadresseerd. Het wegnemen van die root cause is het onderwerp van deze white paper.

Geschiedenis

Sinds ICT in de jaren '60 van de vorige eeuw een vlucht heeft genomen kent ons land een geschiedenis van decentralisaties, reorganisaties, fusies en overnames. Continu zijn we bezig geweest ICT dan weer te ontvlechten, dan weer samen te voegen. We kenden periodes waarin afdelingen ieder hun eigen ICT-oplossingen gingen realiseren. We kenden periodes waarin we die weer in elkaar gingen schuiven. En we kenden ook periodes waarin we het applicatielandschap met dat van andere organisaties integreerden. Het resulteerde in een lappendeken van applicatieplatformen, technologieën en leveranciers.

Daarnaast heeft de technologie allerminst stilgestaan: voortdurend zijn nieuwe inzichten, technieken, tools en platformen geïntroduceerd. De ontwikkeling van internet-technologie heeft de rol van ICT voorgoed veranderd, maar wel met een grote impact op het landschap.

De druk op sneller implementeren, kortere ontwikkelcycli en verlagen van de time-to-market is toegenomen. Het product moet nu eenmaal snel op de markt verschijnen om concurrerend te blijven. Er was weinig aandacht voor architectuur, en er was al helemaal geen focus op het opruimen van oude systemen.

Dit alles heeft geleid tot de situatie waarin we nu zitten: verschillende generaties van applicaties, platformen, processen en tools, een landschap waarin een end-to-end-overzicht ontbreekt, hoge beheer-, onderhouds- en innovatiekosten. Kortom een risico voor de continuïteit van organisaties.

White paper: De sleutel tot het beheersen van complexiteit

Wat is complexiteit?

Voordat een ingrijpend traject wordt gestart met complexiteitsreductie als inzet, moet eerst consensus over het begrip bestaan.

Een definitie van complexiteit is de volgende:

Complexiteit is de mate waarin een proces of systeem begrepen wordt

Bron:

www.encyclo.nl/begrip/complexiteit

Als we deze definitie toepassen op het applicatielandschap, gaat het om begrip van en inzicht in het landschap. Betrokkenen, zoals beheerders, onderhoudsprogrammeurs, architecten, business analisten, gebruikers, et cetera, moeten er hun werk mee kunnen doen.

Om meer inzicht in het begrip te geven is het zinvol onderscheid te maken tussen twee typen complexiteit: onvermijdelijke complexiteit en vermijdbare complexiteit.

We spreken van *onvermijdelijke complexiteit* bij een grote organisatie met bijvoorbeeld veel afdelingen, veel klanten, veel producten, veel applicaties en veel bedieningskanalen. Dan hoort complexiteit er nu eenmaal bij.

Maar sommige vormen van complexiteit staan daar los van, zijn te voorkómen: we spreken dan van *vermijdbare complexiteit*. Hiervan is sprake als we complexiteit bewust of onbewust laten toenemen, terwijl ook ándere keuzes mogelijk waren. Bijvoorbeeld een bewuste keuze voor het redundant opslaan van data om afhankelijkheden met een andere applicatie of afdeling te vermijden of om performance-redenen. Of het goedkeuren van

een ontwerp zonder dat iemand op complexiteitstoename heeft getoetst.

Daarnaast kan onderscheid worden gemaakt tussen functionele en technische complexiteit, omdat deze zich verschillend manifesteren maar ook omdat het eigenaarschap in de organisatie doorgaans verschillend belegd is.

Functionele complexiteit is complexiteit die tijdens *het gebruik* van een applicatie wordt ervaren. Bijvoorbeeld eenzelfde functie die zich op verschillende plaatsen anders gedraagt. Of een situatie waarbij de gebruiker overbodig werk moet doen om iets voor elkaar te krijgen ('workarounds').

Technische complexiteit wordt tijdens *het beheer* ervaren. De gebruiker merkt er niets van. Bijvoorbeeld een functie, tabel, veld et cetera, is op verschillende plaatsen vastgelegd/gedefinieerd, moet daardoor bij onderhoud op evenzoveel plaatsen worden aangepast ('short-cuts'). Hierdoor kosten kleine wijzigingen relatief veel tijd. Soms is zelfs bijvoorbeeld een volledige regressietest nodig, omdat men niet zeker weet in hoeverre de oplossing wel overal is doorgevoerd.

Deze white paper betreft de vermijdbare

| COMPLEXITEIT | Functioneel | Technisch |
|----------------|-------------|-----------|
| Onvermijdelijk | X | X |
| Vermijdbaar | ✓ | ✓ |

Figuur 1: De scope van deze white paper

complexiteit, zowel functioneel als technisch.

De problemen

1. Kosten beheer en onderhoud

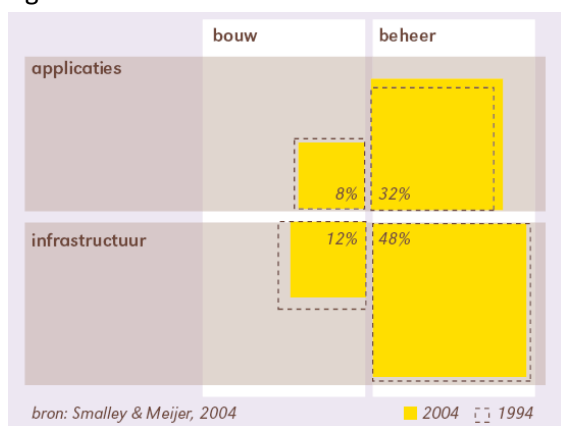
De Wet van toenemende complexiteit van Manny Lehman ¹:

Ofwel: onderhouds- en beheerkosten nemen

As an evolving program is continually changed, its complexity, reflecting deteriorating structure, increases unless work is done to maintain or reduce it

M. Lehman, 1980

toe door toenemende complexiteit. In 2004 was al bekend dat ongeveer 80 procent van de ICT-budgetten besteed werd aan beheer: zie Figuur 2.



Figuur 2: Bouwkosten ↔ beheerkosten

Inmiddels is het cijfer verder opgelopen tot 90 procent van de ICT-budgetten.

Daarnaast heeft onderzoek aangetoond dat het in de lucht houden van een ICT-landschap ieder jaar vijf procent meer kost dan het jaar daarvoor 1. En dan is er nog geen wijziging doorgevoerd. De kosten worden zó hoog dat er nauwelijks meer budget overblijft voor innovatie, ten einde de concurrentie voor te blijven.

¹ Manny Lehman: Proceedings of the IEEE, vol. 68, nr. 9, September 1980, pp. 1068

2. Flexibiliteit

Het is erg moeilijk geworden om nieuwe systemen toe te voegen of uit te faseren, daarnaast is het doorvoeren van een change een tijdrovende en dure zaak geworden. Doordat het totale inzicht in de samenhang tussen de systemen ontbreekt, wordt de flexibiliteit verlaagd en de time-to-market verhoogd. Daarmee is de oorspronkelijke 'enabler' geworden tot een 'disabler'. ICT blokkeert de vooruitgang!

3. Inconsistentie in de bedrijfsvoering

Aangezien meerdere applicaties dezelfde bedrijfsprocessen ondersteunen, al dan niet op verschillende platformen, maar anders functioneren en ook gebruik maken van verschillende gegevens (bijvoorbeeld over klanten) is het bedrijf als geheel inconsistent. Klanten begrijpen niet dat bij het ene loket bekend is dat hij klant is, maar bij het andere niet. Of dat bij het ene loket andere of minder gegevens bekend zijn dan bij het andere. Dit kost klandizie.

4. Workarounds

Door gebrek aan beheersing van complexiteit is een wildgroei aan functionaliteit ontstaan. Identieke functionaliteit gedraagt zich anders of ziet er anders uit in verschillende applicaties, wat voor gebruikers verwarrend is. Oplossen van kleine fouten is dusdanig tijdrovend, dat regelmatig gekozen wordt voor workarounds, waarbij bijvoorbeeld een beheerder namens de gebruiker een veld moet corrigeren of waarbij meerdere applicaties parallel open moeten staan om de gebruiker zijn werk te kunnen laten doen.

5. Short-cuts

Door druk op time-to-market, te weinig focus op onderhoud of gebrek aan inzicht in de gehele applicatieketen wordt naar snelle

White paper: De sleutel tot het beheersen van complexiteit

oplossingen gezocht, waarbij minder afstemmingen nodig zijn. In de hoop en verwachting dat ze tijdens onderhoud wel weer zullen worden opgelost. Maar daar ontbreekt vaak de aandacht voor. Extra complexiteit en onderhouds-inspanningen zijn het gevolg.

Succesfactoren

Wanneer organisaties het complexiteitsvraagstuk blijvend willen oplossen, zijn de volgende succesfactoren van belang:

1. Consensus over het begrip 'complexiteit'
2. Eerst beheersen, daarna reduceren
3. Beschouw de organisatie als een sociaal systeem
4. Diepgaand inzicht in de interne werking van dit systeem is noodzakelijk
5. De strategie is leidend
6. Gebruik geaccepteerde modellen als referentie

1. Consensus over het begrip 'complexiteit'

Eerder in deze white paper is een definitie gegeven van het begrip 'complexiteit' en is het begrip verder genuanceerd. Als een organisatie daadwerkelijk met complexiteitsbeheersing aan de slag wil, moet er eerst consensus bestaan over het begrip en over de na te streven doelstellingen.

Bijvoorbeeld:

- Gaat het om functionele of om technische complexiteit of beide?
- Is de doelstelling kostenbeheersing, gebruikerstevredenheid, time to market, of iets anders?

2. Eerst beheersen, daarna reduceren

Complexiteitsreductie is slechts duurzaam als toekomstige complexiteitstoename wordt

beheerst. En daarin speelt ICT-Governance een majeure, maar in de praktijk tot op heden onderbelichte, rol.

Enkele voorbeelden:

Voorbeeld 1

Een ontwerp wordt goedgekeurd door de opdrachtgever. Maar wat zijn de criteria voor goedkeuring?

Doorgaans wordt een ontwerp beoordeeld op de mate waarin het de requirements realiseert, op de mate waarin het aan (architectuur-)standaarden voldoet en op de mate waarin het logisch correct is. Maar als een organisatie zich ten doel heeft gesteld de complexiteit terug te dringen, zijn op dit vlak ontwerpnormen nodig én moet de verantwoordelijkheid om hierop te toetsen worden belegd.

Voorbeeld 2

Onder tijdsdruk wordt een suboptimaal ontwerp goedgekeurd, wetende dat het ontwerp de vermijdbare complexiteit verhoogt. Wie is ervoor verantwoordelijk om het later op te lossen?

De primaire vraag hier is wie het ontwerp goedkeurt en of die goedkeuring door de juiste partijen plaatsvindt. De toegenomen complexiteit leidt tot extra onderhoudstijd en –kosten. Daarmee komt dit op het bordje van de applicatie-eigenaar. Maar is wel duidelijk wie de applicatie-eigenaar is, hoe die zich verhoudt tot de beheerder en wat zijn verantwoordelijkheden zijn? Dit soort vragen is van belang om complexiteit verantwoord te beheersen.

Door op deze wijze naar complexiteit te kijken, krijgt de opdrachtgever een belangrijk instrument in handen voor beheersing ervan, en kan vervolgens met de reductie worden gestart.

White paper: De sleutel tot het beheersen van complexiteit

3. Beschouw de organisatie als een sociaal systeem

Ieder systeem, zoals de ICT-Governance, bestaat uit elementen met een onderlinge samenhang. Binnen een organisatie én in de relatie met de omgeving vormen de mensen die elementen. De samenhang tussen de elementen is de interactie tussen die individuen.



Zonder mensen en hun onderlinge communicatie en afspraken is er geen organisatie: resultaten, zoals producten en diensten, komen tot stand door communicatie en interactie tussen mensen. Een bakker verkoopt brood omdat er klanten zijn die er om vragen. En de secretaresse maakt notulen in opdracht van haar baas.

4. Diepgaand inzicht in de interne werking van dit systeem is noodzakelijk

Wie een systeem, zoals de ICT-Governance, wil verbeteren, moet diepgaand inzicht hebben in de interne werking, de *constructie*² van het systeem. Onder die constructie verstaan we de *elementaire rollen/functies* en de onderlinge *interacties*. Een goede constructie voldoet aan normen op de volgende twee vlakken:

- Elementaire Interactie
- Elementaire rollen/functies

² Proefschrift Rapid Enterprise Design, J.Mulder, TU Delft, April 2006

Elementaire Interactie: er bestaat een gegeven en bewezen universeel patroon van de manier waarop mensen samenwerken 2.

Onderstaande Norm 1 vat dit patroon samen

Norm 1

Iedere succesvolle interactie kent vijf volgordelijke stappen:

- a. Het verzoek (request)*
- b. De belofte (promise)*
- c. De uitvoering (produce)*
- d. De verklaring (state)*
- e. De acceptatie (accept)*

Een voorbeeld:

Een voorbeeld van een succesvolle interactie
Een opdrachtgever vraagt de Project Manager een plan op te stellen (request). De Project Manager bevestigt wanneer het plan klaar is (promise), stelt het plan op (produce), stuurt het op (state), waarop de opdrachtgever het goedkeurt (accept).

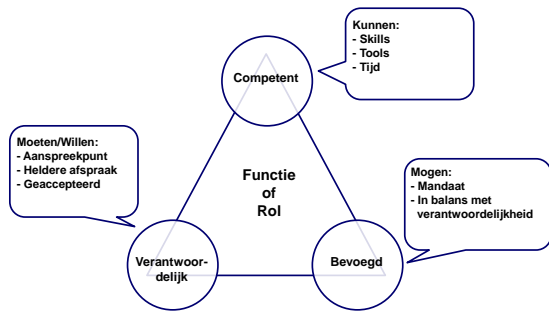
In de praktijk is een veelvoorkomende bron van samenwerkingsproblemen het niet volledig of onjuist doorlopen van deze standaardcyclus. Voorbeelden:

- Er ontbreken interacties. Bijvoorbeeld: er is geen expliciete goedkeuring van een ontwerp door de applicatie-eigenaar
- Er vindt geen acceptatie plaats. Bijvoorbeeld: de Business is opdrachtgever van Technisch Beheer maar laat hen zelf de prioriteiten voor onderhoud bepalen
- Interacties zijn onduidelijk. Bijvoorbeeld: Businessmanagement keurt een architectuur goed zonder dat de concrete consequenties van goedkeuring in kaart zijn gebracht

Elementaire rollen en functies: een rol of functie is een samenhangende verzameling

White paper: De sleutel tot het beheersen van complexiteit

van verantwoordelijkheden, bevoegdheden en competenties, zie Figuur 3.



Figuur 3: Een functie/rol met een gebalanceerde mix van verantwoordelijkheden, bevoegdheden en competenties

En de bijbehorende norm:

Norm 2

Rollen en functies rollen zijn effectief als ze voor iedere interactie een coherente mix van verantwoordelijkheden, bevoegdheden en competenties in zich hebben

Een voorbeeld:

Een voorbeeld: de rol van applicatie-eigenaar

- Verantwoordelijk: aanspreekbaar als een applicatie niet werkt in overeenstemming met de eisen, zowel functioneel (inhoudelijk) als niet-functioneel (conform SLA)
- Bevoegd: zorgt voor funding van onderhoud, bepaalt inhoudelijk de wijzigingsvoorstellen, accepteert wijzigingen
- Competent: begrijpt ICT, kent gebruikerswensen, is in staat de juiste prioriteiten te stellen

Ook bij onvolledige roldefinities ontstaan nogal eens problemen. Voorbeelden:

- Bevoegdheid zonder competenties: de applicatie-eigenaar bepaalt de prioriteiten bij onderhoud, maar heeft geen zicht op complexiteit en de consequenties ervan
- Competentie zonder bevoegdheid: ICT-medewerkers hebben dit zicht wel, maar

er wordt niet naar hen geluisterd bij het stellen van de prioriteiten

- Bevoegdheid zonder verantwoordelijkheid: een applicatie-eigenaar bepaalt de prioriteiten bij onderhoud maar wordt nooit beoordeeld op onderhoudskosten

Bovengenoemde afwijkingen van de normen worden *constructiefouten* genoemd. Iedere organisatie heeft dergelijke constructiefouten, die de organisatie beletten haar optimale performance te halen. Het is daarom waardevol de huidige constructie in kaart te brengen, bijvoorbeeld in de vorm van een constructiediagram. Op basis van gevonden normafwijkingen is de organisatie vervolgens in staat gericht constructieverbeteringen door te voeren, waar die het meest waarde toevoegen.

Zo ontstaat inzicht in:

- De ICT-besluitvormingsmomenten waarop de organisatie complexiteit bewust of onbewust laat groeien, alsmede de oorzaken ervan
- De mate waarin besluitvormers de juiste mix van verantwoordelijkheden, bevoegdheden en competenties hebben voor optimale besluitvorming
- De wijze waarop de huidige ICT-Governance de organisatiestrategie bedreigt
- Haalbare, passende verbetermogelijkheden voor alle bovenstaande punten

5. De strategie is leidend

Voor iedere organisatie is de strategie leidend. Op verschillende momenten is toetsing



White paper: De sleutel tot het beheersen van complexiteit

tegen die strategie en doelstellingen zinvol. Het is daarom van belang om de huidige ICT-Governance-constructie te toetsen aan de strategie en doelstellingen van de organisatie. Het resultaat is inzicht in de belemmeringen die de constructie opwerpt voor het behalen van die strategie en doelstellingen.

Stel bijvoorbeeld dat een postorderbedrijf een consistente klantbeleving hoog in het vaandel heeft staan, waarmee onder andere bedoeld wordt dat alle internetapplicaties eenzelfde 'look and feel' hebben. Als dit niet is geborgd, bijvoorbeeld door uniforme en herhaalbare toetsing op functionele complexiteit, komt dat aspect van de strategie in gevaar.

Ook verbetervoorstellen en implementatie moeten op analoge wijze aan de strategie worden getoetst.

6. Gebruik geaccepteerde modellen als referentie

In veel organisaties worden referentiemodellen (ITIL, BiSL, CMMI, ASL) en/of methodes (Prince2, Scrum, RUP, DSDM) gebruikt ten behoeve van kwalitatief goede ICT-processen. Waar dit het geval is, doet een organisatie er goed aan om consistentie tussen de ICT-Governance-constructie en die modellen en methodes te toetsen en van eventuele afwijkingen de impact te bepalen. Als bijvoorbeeld een Prince2 Senior Supplier of een Scrum Product Owner de achterban niet optimaal vertegenwoordigt doordat er onvoldoende mandaat is, komt succes van projecten in gevaar.

Conclusie

Randvoorwaardelijk voor duurzame complexiteitsreductie is het onder controle hebben van de complexiteitstoename. ICT-Governance is hierbij een majeure factor, daarom is het zinvol om deze in detail in kaart te brengen. Een effectieve manier daarvoor is

het beschouwen van de ICT-Governance als een sociaal systeem, dat wil zeggen door het te beschouwen zien als een netwerk van elementaire rollen die met elkaar interacteren. Het toetsen van deze ICT-Governance aan een objectief normenkader, aan de strategie en aan gebruikte modellen en methoden geeft diepgaand inzicht in fouten in de constructie. Daarmee creëert de organisatie handvatten om de ICT-Governance blijvend te verbeteren en de complexiteit te beheersen.

De auteur

Peter Brouwer is organisatieadviseur en verandermanager bij Delta Change Consultants. Zijn kracht zit hem in het implementeren van pragmatische en duurzame oplossingen op basis van zijn jarenlange ervaring met organisatieadvies op diverse terreinen.

Feedback

U bent van harte uitgenodigd tot het geven van feedback op een van de volgende emailadressen: theo.janssen@deltacc.nl of peter.brouwer@deltacc.nl.