



Configuratiebeheer – wijzigingenbeheer – releasebeheer

De niet-Bermuda-driehoek!

Iedereen kent de beruchte Bermuda-driehoek. Er raakt van alles in zoek. ITIL en ASL kennen de driehoek configuratiebeheer – wijzigingenbeheer – releasebeheer. Anders dan de Bermuda-driehoek staat deze er garant voor dat juist helemaal niets zoekraakt. Deze 'de niet-Bermuda-driehoek!' verdient het om ook te worden ingevoerd in de domeinen business en functioneel beheer, zo luidt de boodschap. Het staat garant voor stabiele businessprocessen, aldus de auteurs... En was dát niet dé voorwaarde voor een goede business IT alignment?

De Bermuda-driehoek ligt tussen Miami, Bermuda en San Juan. Vroeger verdwenen daar vaak vliegtuigen en schepen op raadselachtige wijze. Sinds de transparantie van scheepvaart- en vliegbewegingen is toegenomen (radar), raakt er eigenlijk nooit meer iets kwijt. Bij technisch beheer en bij applicatiebeheer is de transparantie van de bewegingen van veranderingen – en vooral veranderingcomponenten – zo groot dat er nauwelijks nog iets kwijt raakt of per ongeluk in de verkeerde omgeving blijft hangen. In de domeinen business en functioneel beheer is dat echter anders. Daar raakt nog wél eens wat zoek, wordt nog wél eens het verkeerde in productie genomen en wordt nog wél eens iets vergeten. Het verschil tussen technisch beheer en applicatiebeheer enerzijds en de domeinen business en functioneel beheer anderzijds is het bestaan van de driehoek configuratiebeheer – wijzigingenbeheer – releasebeheer.

Anders dan de Bermuda-driehoek staat deze driehoek er garant voor dat juist helemaal niets zoekraakt of per ongeluk juist wel of juist niet in een verandering wordt meegenomen. Deze 'niet-Bermuda-driehoek' verdient het om te worden ingevoerd in de domeinen business en functioneel beheer, zodat ook daar alles in één keer goed gaat.

Vastlopende businessprocessen

In de meeste SAP-omgevingen worden de parameters behandeld alsof het programmatuur is: parameterwijzigingen volgen het wijzigingsproces, worden goed getest en worden door beheer met gepaste argwaan benaderd. En terecht: parameterwijzigingen kunnen grote, ongewenste gevolgen hebben. De meeste 'gewone' applicaties, of het nu maatwerk of standaardpakketten betreft, hebben ook vele parameters, die vaak zomaar worden veranderd, zonder bovenstaande 'verworvenheden'. Soms met desastreuze gevolgen, zoals onjuiste functionaliteit, vastlopende programmatuur en daardoor vastlopende businessprocessen. Ook de uitgifte van verkeerde handleidin-

heid. Configuratiebeheer definieert de objecten van beheer. Dat zijn de objecten waar we naar kijken. Niet meer, niet minder. Wijzigingenbeheer zorgt er voor dat we het erover eens zijn wat we willen veranderen en welke risico's we daarbij lopen. Releasebeheer ten slotte staat garant dat we precies de beoogde wijzigingen doorvoeren. Niet meer en niet minder.

Een verandering is een wijziging – in de betekenis van ITIL – als die leidt tot een verandering in de configuratiedatabase (CMDB). Dat maakt het ontwerp van die database (welke soort items neem ik wel of niet op in die database) bepalend voor de scope van wijzigingenbeheer en in het verlengde daarvan van releasebeheer. Als ik er

Met behulp van transparantie wordt nog meer transparantie afgedwongen

gen, het per ongeluk niet implementeren van businessproces-aanpassingen en andere aanpassingen in het businessdomein kunnen leiden tot ernstige verstoringen in de business.

ITIL-driehoek

Binnen ITIL zorgt een gedegen driehoek voor de benodigde transparantie en zeker-

voor kies om toetsenborden op te slaan in de CMDB, dan moet een nieuw toetsenbord door het wijzigingsproces en zullen we nieuwe toetsenborden in een release ook laten lopen via releasemanagement. Als ik daar niet voor kies, gaan die toetsenborden buiten het wijzigingsproces om. Configuratiebeheer draagt bij aan transparantie, omdat het zorgt voor een exacte

weergave van de werkelijkheid.

In wijzigingenbeheer worden risico's afgewogen, opdat de business keuzes kan maken. Deze risico's zijn:

- ♦ Welke beoogde wijzigingen kunnen elkaar beïnvloeden en welke risico's lopen we bij gezamenlijke implementatie?
- ♦ Welke continuïteitsrisico's lopen we bij de implementatie van een wijziging?
- ♦ Welke zekerheden hebben we ingebouwd?
- ♦ Welke maatregelen kunnen we nemen?

Eenvoudige (standaard) wijzigingen doorlopen een andere procedure (eenmalige toetsing van de soort), net als wijzigingen ten behoeve van incidentbestrijding (toetsing achteraf in plaats van vooraf). In de kern echter gebeurt steeds hetzelfde. Wijzigingenbeheer draagt bij aan transparantie, omdat het de beoogde veranderingen in kaart brengt, samen met de risico's die eraan verbonden zijn en omdat het de beslissingen afdwingt en vastlegt ten aanzien van die veranderingen.

Releasebeheer zorgt voor de uitvoering van de wijzigingen door beheer. Releasebeheer beheert de fysieke voorraad configuratie-items, stelt de te veranderen configuratie-items ter beschikking en neemt de veranderingen in beheer die zijn goedgekeurd binnen wijzigingenbeheer. De veranderaars moeten precies aangeven welke veranderingen er allemaal in de release zitten, want iemand anders voert de veranderingen door namens beheer. Improvisatie is niet mogelijk. Releasebeheer draagt bij aan transparantie, omdat het wijzigingen clustert in overzichtelijke releases en omdat het zorgvuldigheid afdwingt van de veranderaars.

Dekkingsgraad driehoek

ITIL en ASL kennen de drie processen, al heet releasebeheer in ASL 'programmabeheer en distributie'. BiSL kent wel wijzigingenbeheer en releasebeheer (transitie), maar die bemoeien zich vooral met IT-zaken en juist niet met de zaken binnen het functioneel beheerdomein. Maar ja, BiSL kent geen configuratiebeheer, dus waar kunnen wijzigingenbeheer en releasebeheer anders over gaan dan over de items van anderen? In het businessdomein zijn ons geen modellen bekend, die ook maar één van de drie processen ondersteunen. Al met al worden de drie processen niet uitgevoerd binnen de domeinen business en functioneel beheer. Dat kan twee dingen beteke-



**In de domeinen
business en functioneel
beheer raakt nog wel
eens wat zoek**

nen. Óf binnen functioneel beheer en business worden de drie processen niet uitgevoerd, omdat er geen modellen zijn die dit afdwingen, óf er zijn binnen business en functioneel beheer geen modellen met deze drie processen, omdat binnen die domeinen niet zo wordt gedacht. Waarschijnlijk zijn ze allebei waar.

Funcitiescheiding

We kennen een (bank)bedrijf waar elke wijziging twee weken in quarantaine gaat. Als je in die periode nog een wijziging hebt, begint de quarantaineperiode weer

opnieuw. Als veranderaar denk je dan wel heel goed na voor je een nieuwe release aanbiedt. En dat is precies wat je wilt bereiken. Bij business en functioneel beheer is het de gewoonte dat de veranderaar zelf alle veranderingen doorvoert. Die heeft daarmee de mogelijkheid om ter plekke te improviseren en daar maakt hij gebruik van ook. Juist de functiescheiding tussen beheeren en veranderen zorgt er voor dat de veranderaar de verandering zorgvuldig plant en dat de beheerder die zorgvuldig en volgens plan uitvoert; ook bij business en functioneel beheer.

Advertentie Topdesk

Functioneel beheer en business hebben recht op dezelfde 'niet-Bermuda-driehoek', op dezelfde zekerheid en dezelfde transparantie als IT al jaren heeft. Ze hebben dus de plicht om serieus werk te maken van de drie processen.

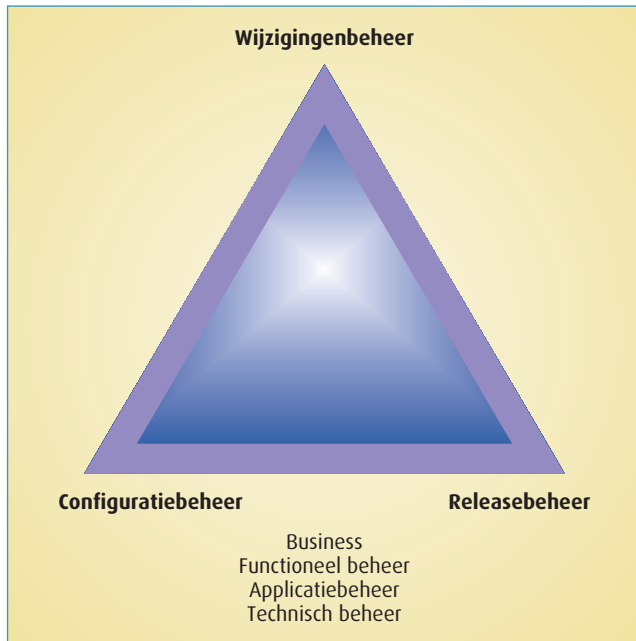
We lopen ze langs:

- ♦ **Configuratiebeheer.** Functioneel beheer moet zich afvragen welke items het in beheer heeft: gegevensstromen, gegevensverzamelingen, parameters, handleidingen, opleidingen, (impact)analyses, strategiedocumenten en nog veel meer. Ook business moet zich een dergelijk vraag stellen: procesgang, procesdocumentatie, functiebeschrijvingen, enzovoort. Loop bijvoorbeeld 'COPAFIJTH' maar door.
- ♦ **Wijzigingenbeheer.** Wijzigingen van configuratie-items moeten op een nette manier door het wijzigingenproces worden gehaald. Per release willen we weten welke risico's we kunnen onderkennen en welke maatregelen we hebben genomen. Per release stellen we vast of we bereid zijn die risico's te nemen.
- ♦ **Releasebeheer.** De configuratie-items zijn in beheer van releasemanagement. Dat stelt de gevraagde items ter beschikking aan veranderaars en het neemt de goedgekeurde releases in beheer namens beheer.

Rolverdeling

Configuratiebeheer kan de verschillende domeinen voor een groot deel onafhankelijk van elkaar doorvoeren. Omdat er uiteraard ook koppelingen bestaan die moeten worden vastgelegd tussen items uit verschil-

De 'niet-Bermuda-driehoek': een evenwichtige opbouw van de drie processen 'configuratiebeheer', 'wijzigingenbeheer' en 'releasebeheer' in alle vier domeinen is dé garantie voor stabiele businessprocessen.



Er is maar één wijzigingenproces dat alle wijzigingen binnen de vier domeinen (business, functioneel beheer, applicatiebeheer en technisch beheer) bestuurt, op basis van de totale configuratiedatabase. Het ITIL-proces beschrijft de rolverdeling tussen de vier domeinen uitstekend en er zijn geen zwaarwegende argumenten om de bestaande praktijk (technisch beheer is proceseigenaar) te veranderen. Maar anders dan bij ITIL moeten we voortaan ook de veranderingen in de domeinen business en functioneel beheer meenemen in de releases en in de discussies.

in één proces door beheer worden doorgevoerd. Het ligt voor de hand om functioneel beheer hier de rol van hoofdaannemer of regievoerder te geven, omdat het primair gaat om businessveranderingen en informatievoorziening de bindende factor is. Elk van de drie processen levert transparantie. Configuratiebeheer maakt duidelijk wat we allemaal hebben, releasebeheer wat we willen veranderen en wijzigingenbeheer wat we daarvan vinden. Maar in de driehoek versterken die transparanties elkaar. De driehoek geeft alle spelers een belangrijke, eigen rol en de driehoek dwingt af dat ze eisen stellen aan elkaar. Met behulp van transparantie wordt nog meer transparantie afgedwongen, zodat er helemaal niets meer kan verdwijnen.

Conclusie

De driehoek garandeert stabiele veranderingen, waarbij er onderweg niets zoek raakt. Een evenwichtige opbouw van de drie processen in alle vier domeinen is de garantie voor stabiele businessprocessen en daarmee voor het succes van de business. De onderneming heeft er recht op.

De driehoek garandeert stabiele veranderingen, waarbij er onderweg niets zoek raakt

lende domeinen, is er wel degelijk een soort hoofdaannemer nodig. De hoofdaannemer is een partij die eisen stelt aan de verschillende domeinen en garant staat dat alle partijen onderling consistent, en waar nodig in samenwerking, hun bijdrage leveren. Het ligt voor de hand die rol te geven aan technisch beheer op basis van het ITIL-proces, dat immers het best is uitgewerkt. Daarbij moet wel expliciet en specifiek rekening worden gehouden met de eigenaardigheden van elk domein.

De verschillende domeinen kunnen elk hun eigen veranderingen doorvoeren, zolang we er maar voor zorgen dat degene die de verandering wil (de veranderaar) niet degene is die de daadwerkelijke verandering doorvoert (beheer). Net als bij het bedenken, ontwerpen en realiseren van de verandering (projectleider), hebben we ook bij het in beheer nemen van de release een hoofdaannemer nodig. Iemand die zorgt dat de verschillende veranderingen in de verschillende domeinen in samenhang en volgens plan

MAURICE KLEIN, JAN REMMERS en BART STOFBERG zijn als consultant werkzaam bij LogicaCMG.