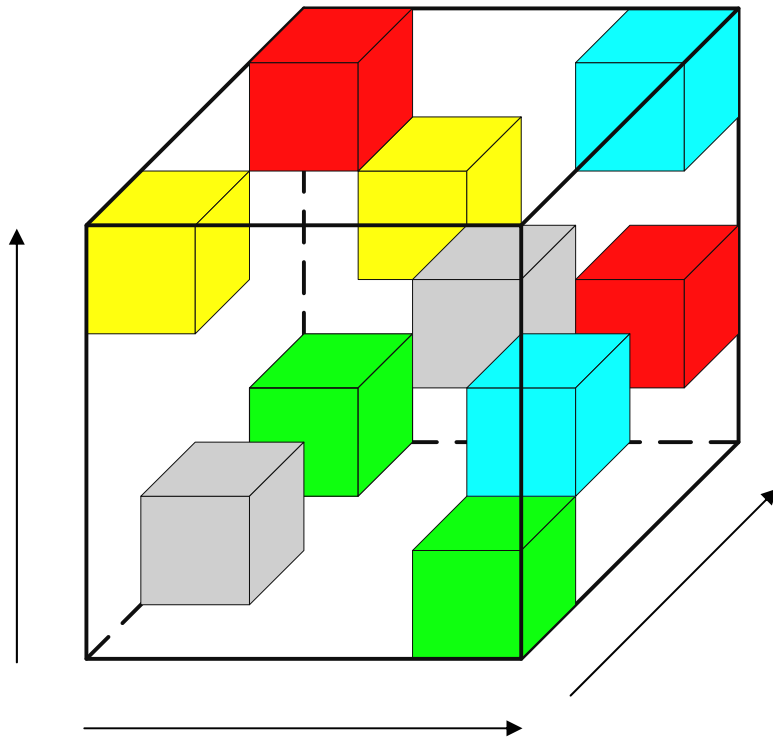


# DE TESTKUBUS

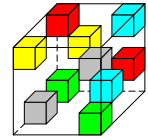
Aanpak voor het structureren en beheren van testsets



Auteur	Expertisegroep TestKubus
Versie	1.0 (25 april 2005)
Plaats	Diemen
Kenmerk	Definitief

TestKubus

Versieinformatie

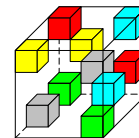


## VERSIEINFORMATIE

Versie	Datum	Bijzonderheden	Auteur
1.0	25 april 2005	Puntjes op de i. Definitieve versie.	Expertisegroep TestKubus

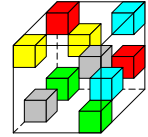
TestKubus

Verzendlijst



## VERZENDLIJST

Dit document is bedoeld voor professionals en management van Sogeti en voor geïnteresseerde klanten. Het wordt verspreid via de kennisbank van Sogeti, onderdeel SC.



## WOORD VOORAF

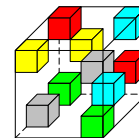
De TestKubus is een aanpak voor het beheren van testsets die in de praktijk is ontwikkeld en voor het eerst door Wout Komduur (relatiemanager) en Paul von Winckelmann (testmanager) is toegepast bij een klant van Sogeti.

Om de jaren van opgedane kennis en ervaring beschikbaar te stellen voor medewerkers van Software Control en overige geïnteresseerden, is een expertisegroep opgericht van ervaren TestKubus-gebruikers. De doelstelling was het verder uitontwikkelen van de TestKubus tot een generieke aanpak die succesvol toegepast kan worden bij al onze klanten.

Deze expertisegroep heeft dankbaar gebruik gemaakt van Arno Verweij en Jack van de Corput, die onbekend waren met de TestKubus-aanpak en als klankbord dienden. Arno heeft tevens de redactie van dit document voor zijn rekening genomen.

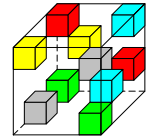
Dit document en de erin beschreven aanpak is het gezamenlijke resultaat van de expertisegroep TestKubus. Aan de totstandkoming hebben de volgende mensen bijgedragen:

Hans Delprat  
Robert Jelgerhuis  
Michel Suiveer  
Arno Verweij  
Guido Wester  
Paul von Winckelmann



## INHOUDSOPGAVE

1.	INLEIDING .....	1
2.	HET CONCEPT VAN DE TESTKUBUS .....	2
3.	DE TESTKUBUS EN BUSINESS DRIVEN TESTMANAGEMENT .....	5
3.1	Risk Based Testing .....	5
3.2	Business Driven Test Management.....	5
4.	HET INRICHTEN VAN DE TESTKUBUS .....	7
4.1	Fase Planning & Beheer .....	7
4.1.1	<i>Voor de fixatie van het testplan.....</i>	<i>7</i>
4.1.2	<i>Na de fixatie van het testplan .....</i>	<i>8</i>
4.2	Fase Voorbereiding .....	8
4.3	Fase Specificatie .....	8
4.4	Fase Uitvoering .....	9
4.5	Fase Afronding.....	9
4.6	Richtlijnen voor het gebruik van zwaartecategorieën .....	9
4.6.1	<i>Bepalen zwaartecategorieën .....</i>	<i>10</i>
4.6.2	<i>Bepalen verdeelsleutel per zwaartecategorie .....</i>	<i>10</i>
4.6.3	<i>Verdelen testgevallen over zwaartecategorieën .....</i>	<i>11</i>
4.7	TestKubus bij een bestaande testset .....	12
5.	HET GEBRUIK VAN DE TESTKUBUS IN ONDERHOUD .....	13
5.1	Fase Planning & Beheer .....	13
5.2	Fase Voorbereiding .....	14
5.3	Fase Specificatie .....	14
5.4	Fase Uitvoering .....	14
5.5	Fase Afronding.....	14
6.	RAPPORTAGE & METRICS .....	15
6.1	Plannen en begroten .....	15
6.2	Rapportage.....	15
7.	HULPMIDDELEN .....	17
7.1	Spreadsheets.....	17
7.2	Testspecificatietools .....	18
8.	DE TESTKUBUS IN DE PRAKTIJK .....	19
8.1	Uitgangssituatie .....	19
8.2	Oplossingen .....	20
8.3	Resultaten.....	20



## 1. INLEIDING

Een goede testset is van onschatbare waarde. Tijdens het realiseren van een applicatie is ze de basis voor de rapportage over testdekking en voortgang. In een beheersituatie is ze een archief van kennis over de toepassing en een bron van kant en klare testgevallen voor regressietests.

In de praktijk zijn testsets vaak gebrekkig. In nieuwbouwprojecten valt het aanvankelijk wel mee, maar naarmate het project tegen het einde loopt neemt de tijdsdruk toe, verschuift de projectfocus geheel naar de op te leveren toepassing en verzandt het onderhoud op de testset. Tegen de tijd dat toepassing en testset in beheer worden genomen, is de discrepantie tussen beiden zo groot dat de testset voor de beheerorganisatie nog maar matig bruikbaar is.

Het opzetten van een doordachte testset is eenvoudiger gezegd dan gedaan. Testers zelf zien de testset meestal ook slechts als een product dat vanzelf ontstaat tijdens het specificeren van testgevallen. Zij zijn niet gewend om de testset in zijn geheel als projectdeliverable te beschouwen en er bijvoorbeeld vooraf acceptatiecriteria voor op te stellen.

In dit document wordt een aanpak beschreven voor het opbouwen, gebruiken en beheren van testsets: de TestKubus. De TestKubus is een samenhangende verzameling van principes waarmee het mogelijk wordt om:

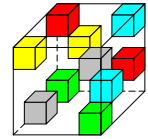
- testgevallen te specificeren en uit te voeren op basis van prioritering;
- snel en adequaat te rapporteren over de voortgang van testspecificatie en/of testuitvoering;
- testtrajecten nauwkeurig te plannen en te begroten;
- snel en variabel regressietestsets samen te stellen;
- wijzigingen in het testobject eenvoudig te verwerken in de testset.

De TestKubus is een toepassing van TMap waarbij één testproduct wordt uitgelicht: de testset. De TestKubus vervangt het gangbare maar onterechte idee van de testset als toevallige verzameling testgevallen door dat van een concreet product: een projectdeliverable die voldoet aan vooraf opgestelde acceptatiecriteria.

Door deze benadering krijgt de (toekomstige) beheerorganisatie meer grip op deze voor hen zo belangrijke projectdeliverable. De TestKubus helpt de kapitaalvernietiging te voorkomen die optreedt wanneer een testset, waaraan vele handen maandenlang hebben gewerkt, bij overdracht naar beheer niet meer bruikbaar blijkt.

De principes toegepast in de TestKubus maken haar zeer geschikt voor gebruik in combinatie met Business Driven Testmanagement (BDTM).

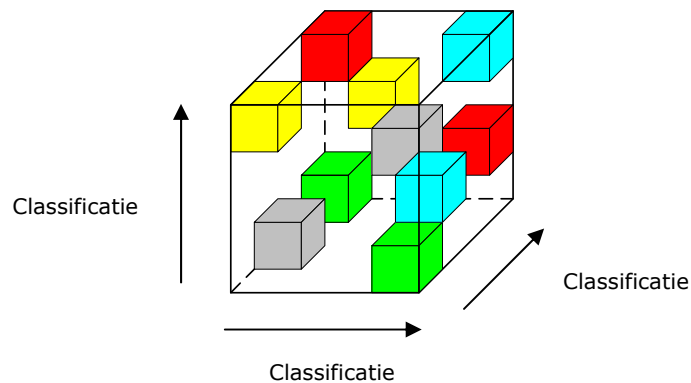
Dit document begint met een beknopte beschrijving van het mechanisme van de TestKubus: de principes erachter, hoe het werkt en wat je ermee kunt. Vervolgens wordt ingegaan op de relatie tussen de TestKubus en BDTM. Het opzetten en gebruiken van de TestKubus wordt per TMap-fase uitgewerkt. Een apart hoofdstuk is gewijd aan rapporteren en begroten met de TestKubus. Het document besluit met praktische hoofdstukken over bruikbare hulpmiddelen en praktijkresultaten.



## 2. HET CONCEPT VAN DE TESTKUBUS

De TestKubus is in concept weinig meer dan een verzameling aanvullende gegevens die per testgeval worden vastgelegd: de testgevallen in de testset worden 'geclassificeerd'. Met behulp van deze classificaties kunnen langs allerlei dwarsdoorsneden subsets van testgevallen uit de testset worden geselecteerd.

De onderstaande figuur verbeeldt de TestKubus. Ieder blokje vertegenwoordigt een testgeval met zijn classificaties.

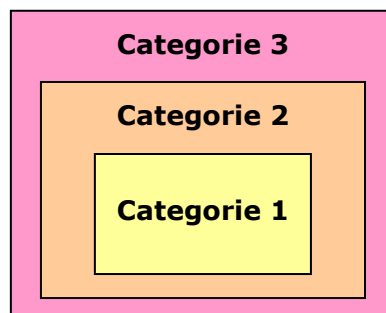


De TestKubus is de vertaling van een selectiemechanisme naar een ruimtelijke figuur, een kubus. Echter, waar een kubus slechts drie dimensies heeft, heeft de TestKubus in principe een onbeperkt aantal selectiemogelijkheden (classificaties) en daarmee oneindig veel dimensies. Deze classificaties kunnen bijvoorbeeld zijn:

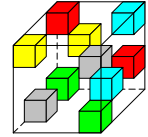
- applicatie
- deelobject
- functie
- productrisico
- risicoklasse
- proces(onderdeel)
- release
- requirement
- transactie
- zwaartecategorie

Een goede selectie van classificaties en het correct classificeren van de testgevallen is bepalend voor de bruikbaarheid van de TestKubus.

Essentieel in de TestKubus is de classificatie naar zwaartecategorie. Deze classificatie geeft het 'gewicht' van het testgeval in de testset aan. Met name deze classificatie maakt het mogelijk om met een variabele diepgang risicogebaseerd te (regressie)testen. Dit werkt als volgt, bijvoorbeeld bij drie zwaartecategorieën:



Zoals de figuur aangeeft, zijn de verschillende zwaartecategorieën deelverzamelingen van elkaar (categorie 3 bestaat uit alle testgevallen van categorie 1, 2 en 3;



categorie 2 bestaat uit de testgevallen van categorie 1 en 2). Categorie 1 bevat de meest elementaire testgevallen, want die maken deel uit van alle overige categorieën.

Elk deelobject heeft testgevallen in elke categorie. De getalsverhoudingen tussen de verschillende categorieën liggen binnen in de praktijk bepaalde marges (bijvoorbeeld categorie 1 = 10-15% van het totaal aantal testgevallen, categorie 2 = 60-70% en categorie 3 = 100%, zie par. 4.6).

De toepassing van deze zwaartecategorieën, bijvoorbeeld bij het samenstellen van regressietestsets, is als volgt:

- Door van een deelobject alleen de testgevallen van categorie 1 te selecteren ontstaat een kleine regressietestset. Deze subset wordt gebruikt voor een deelobject waarin geen aanpassingen zijn gedaan (of voor een intaketest op een nieuw of ingrijpend gewijzigd deelobject).
- De testgevallen van categorie 2 (= inclusief categorie 1) leveren een normale regressietestset op, bijvoorbeeld voor een deelobject waarin aanpassingen zijn gedaan.
- De testgevallen van categorie 3 (= inclusief categorie 1 en 2) dekken het totale deelobject af en worden toegepast bij nieuwe of ingrijpende gewijzigde deelobjecten.

De zwaartecategorieën kunnen ook voor andere doeleinden gebruikt worden. Zie daarvoor de hoofdstukken voor het inrichten en gebruik van de TestKubus.

Het kunnen segmenteren van de testset met behulp van deze zwaartecategorieën is essentieel voor de TestKubus. Het is voorwaardelijk voor de schaalbaarheid van de (regressie)test en mag in geen geval achterwege blijven. In paragraaf 4.6 worden richtlijnen gegeven hoe testgevallen over zwaartecategorieën te verdelen.

De TestKubus schrijft geen vaste testspecificatietechnieken voor. Alle bestaande technieken kunnen worden gebruikt om de testgevallen in de testset te specificeren. De keuze voor een techniek wordt op de normale manier bepaald aan de hand van de teststrategie.

Ook aan de mate van detail waarin de testgevallen gespecificeerd worden, stelt de TestKubus geen eisen. Wanneer wordt voorzien dat testgevallen uitgevoerd gaan worden door testers zonder materiekennis, kunnen de testgevallen meer in detail worden uitgeschreven. Deze keuze is voornamelijk afhankelijk van de organisatie van het testtraject.

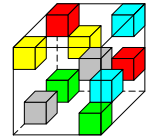
Slechts op één punt stelt de TestKubus een eigen specifieke eis aan de testgevallen: ze moeten onafhankelijk van elkaar zijn. Dit is het zogenaamde onafhankelijkheidsprincipe in de TestKubus:

- Er bestaan geen afhankelijkheden tussen testgevallen.  
Wanneer de uitvoering van het ene testgeval voorwaardelijk is voor de uitvoering van het andere, ontstaan problemen bij het samenstellen van testsets uit de testkubus: het kan zomaar zo zijn dat er eerst allerlei testgevallen uitgevoerd moeten worden die niet tot de eigenlijke testset behoren.
- Testgevallen kunnen parallel aan elkaar worden uitgevoerd.  
Testgevallen die voor een bepaalde periode exclusief gebruik van de testomgeving vereisen, belemmeren de uitvoering van andere testgevallen. Dit bemoeilijkt het plannen van de doorlooptijd van het testtraject.

Bij toepassing van de TestKubus worden de omvang van de testset en de daarmee samenhangende activiteiten in het testtraject concreet meetbaar. Door per testgeval

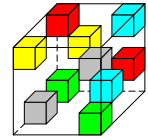


TestKubus



Het concept van de TestKubus

enkele statussen bij te houden, biedt de TestKubus op elk moment een actueel beeld van de voortgang. Wanneer metrics bijgehouden worden van afgeronde of lopende testtrajecten, kan met de TestKubus een nauwkeurige planning opgesteld worden voor toekomstige trajecten of voor het vervolg van lopende activiteiten.



## 3. DE TESTKUBUS EN BUSINESS DRIVEN TESTMANAGEMENT

### 3.1 Risk Based Testing

Toepassing van de Testkubus is de ideale manier om invulling te geven aan Risk Based Testing. De testset in de TestKubus is het product van een teststrategie op grond waarvan sommige deelobjecten met meer diepgang getest worden dan andere. In de risicoanalyse die aan die teststrategie ten grondslag ligt, wordt uitgegaan van de elementaire risicofactoren *foutkans*, *gebruiksfrequentie* en *schade*: Risk Based Testing.

Bij een applicatie in onderhoud wijzigt er doorgaans weinig aan de risicofactoren *gebruiksfrequentie* en *schade*. Door de aanpassingen neemt vooral de *foutkans* toe. In de teststrategie voor een onderhoudsrelease kan worden bepaald voor welke deelobjecten een normale of verhoogde *foutkans* geldt. Met behulp van de TestKubus kan dan eenvoudig per deelobject een meer of minder uitgebreide regressietestset worden samengesteld: Risk Based Regression Testing.

Een belangrijke randvoorwaarde voor Risk Based (Regression) Testing is dat de testset in de TestKubus *een goede afspiegeling is van de huidige applicatie*. Wordt niet aan deze voorwaarde voldaan, dan is een regressietestset van beperkte waarde: een door risico's gemotiveerd uittreksel uit een testset waarvan niet precies duidelijk is welke risico's die zelf afdekt.

Mocht het zo zijn dat als gevolg van een onderhoudsrelease ook de risicofactoren *gebruiksfrequentie* en/of *schade* significant wijzigen, dan kan dit tot gevolg hebben dat de testset structureel moet worden aangepast.

### 3.2 Business Driven Test Management

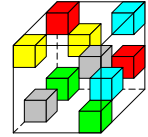
Business Driven Testmanagement (BDTM) is een testaanpak die de business in staat stelt om een testproces aan te sturen op kosten en risico's. Hiervoor is informatie nodig uit het testproces. De TestKubus biedt deze informatie. Dit wordt toegelicht aan de hand van drie van de voor BDTM essentiële elementen.

#### Teststrategie

De TestKubus maakt het mogelijk (met name in onderhoudssituaties) om al bij het bepalen van de teststrategie de kosten van het afdekken van een risico concreet inzichtelijk te hebben. Met een goed ingerichte TestKubus, inclusief betrouwbare kengetallen, kan eenvoudig bepaald worden welke testgevallen een risico of deelobject met welke zwaartecategorie afdekken en wat het kost om deze testgevallen uit te voeren. Deze kosten kunnen concreet worden afgezet tegen de kosten die zouden optreden in het geval dat het risico zich manifesteert. Het resultaat hiervan is een teststrategie waarin door de business bewustere keuzes zijn gemaakt voor het nemen of afdekken van bepaalde risico's.

#### Begroten en plannen

Op basis van de teststrategie wordt de begroting opgesteld. Indien er voor de applicatie een TestKubus is ingericht en betrouwbare ervaringscijfers beschikbaar zijn (in onderhoudssituaties) kan zeer nauwkeurig begroot worden. In nieuwbouwtrajecten is het voordeel minder groot. Maar omdat al vroegtijdig over de omvang van de testset wordt nagedacht, helpt de TestKubus ook hier bij het begroten en plannen. Al was het maar bij het bijstellen daarvan gedurende het testtraject.

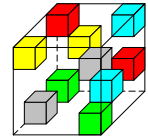


Het gebeurt in de praktijk regelmatig dat een testbegroting in de ogen van de opdrachtgever te hoog uitvalt. Dit wordt dan opgelost door aan de hand van de oorspronkelijke teststrategie van elk deelobject wat testtijd af te halen. Het testtraject wordt daarmee over de gehele linie dunner, zonder dat men inzicht heeft in de extra risico's die men daarbij loopt. Toepassing van de TestKubus verschaft wel inzicht in de (kosten voor het afdekken van) risico's en stelt de opdrachtgever in staat zelf aan te geven op welke punten hij meer risico's wil lopen. Op basis hiervan worden de teststrategie en de testset aangepast om een verlaging van het aantal benodigde uren te realiseren.

### **Rapporteren en sturen**

Het binnen tijd en budget uitvoeren van het testproces is belangrijke informatie voor de business. Met behulp van de TestKubus kan aan de business gedurende het gehele testtraject periodiek of ad hoc inzicht verschaft worden in de kwaliteit van het testobject, de openstaande risico's en de voortgang van het testproces. Afhankelijk van de gekozen hulpmiddelen kan snel gerapporteerd en bijgestuurd worden. Ook in deze situatie kan de besparing van het nemen van extra risico's ogenblikkelijk in tijd en geld worden uitgedrukt. Dit stelt de business in staat om tijdig maatregelen te nemen en, indien nodig, het project en/of het testtraject bij te sturen. Ook in deze situatie kan de besparing van het nemen van extra risico's of de kosten van het inzetten van extra resources ogenblikkelijk in tijd en geld worden uitgedrukt.

Rapportage naar de stakeholders aan de businesszijde in termen van risico's en de afdekking daarvan wordt zeer eenvoudig indien productrisico's of risicoklassen als classificatie in de TestKubus zijn opgenomen. Vanuit dezelfde TestKubus kan naar andere stakeholders langs andere classificaties worden gerapporteerd.



## 4. HET INRICHTEN VAN DE TESTKUBUS

In dit hoofdstuk wordt de inrichting van de TestKubus beschreven met als uitgangspunt de situatie dat er nog geen testset bestaat, zoals aan het begin van een project. De stappen zijn gerelateerd aan de TMap-fasering. De activiteiten gericht op het opzetten van de TestKubus zijn additioneel aan de activiteiten die TMap voor de verschillende fasen onderkent.

In paragraaf 4.7 wordt beschreven hoe een reeds bestaande testset in een TestKubus ondergebracht kan worden.

### 4.1 Fase Planning & Beheer

#### 4.1.1 Voor de fixatie van het testplan

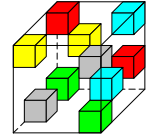
In de eerste plaats moet worden onderzocht of het haalbaar is om de testset voor het project in een TestKubus onder te brengen en zo ja, welke requirements er gelden voor de testgevallen en de TestKubus.

De TestKubus is een optionele aanvulling op TMap. Tegenover de voordelen in het gebruik van de TestKubus, staan investeringen die gedaan moeten worden. Tevens stelt de toepassing van de TestKubus eisen aan de structurering van het testproces. Diverse argumenten kunnen de keuze voor een TestKubus positief (+) of negatief (-) beïnvloeden:

- (+) De opdrachtgever stelt expliciete eisen aan de testset, bijvoorbeeld testen op basis van onderkende risico's (zie hoofdstuk 8 voor een praktijkvoorbeeld).
- (+) Periodieke releases en incrementele ontwikkelmethoden vereisen regelmatig onderhoud op de testset en frequente regressietesten, hetgeen de TestKubus goed ondersteunt.
- (+) Onder tijdsdruk is er inzicht in en grip op het testtraject nodig en tevens een goed beargumenteerde afbakening van de testset.
- (-) Bij eenmalige tests (bijvoorbeeld testen van conversies) is het rendement van de TestKubus laag (de winst komt bij regelmatig testen). Al biedt de TestKubus ook bij eenmalige testtrajecten het voordeel van een goed inzicht in de testset.
- (-) Onvolwassenheid in de test- of projectorganisatie maken de investeringen in de beheersing en structurering van het testproces hoog en verhogen tevens het risico op lage baten door onervarenheid van testorganisatie of projectmanagement.
- (-) De testorganisatie kent geen dedicated testteamleden.

De belangrijkste requirements voor de testgevallen in de TestKubus en voor de TestKubus zelf kunnen worden afgeleid uit de teststrategie. Aanvullende requirements betreffen de manier waarop de testgevallen gespecificeerd worden. Sommige requirements hangen specifiek samen met het gebruik van de TestKubus, andere gelden voor elk testtraject:

- Welke deelobjecten worden onderkend?
- Welke productrisico's dekt het testtraject af (indien met BDTM wordt gewerkt)?
- Hoeveel zwaartecategorieën zijn er nodig en wat is de procentuele verdeling van testgevallen over deze categorieën (zie hoofdstuk 2 voor ervaringscijfers).



- Welke overige classificaties zijn nodig? Elke classificatie voegt een dimensie toe aan de TestKubus, maar dit mag niet ten koste gaan van de inzichtelijkheid en werkbaarheid.
- Op welk niveau moeten de testgevallen beschreven worden: logisch, fysiek of allebei (worden ze uitgevoerd door testers met of zonder materiekennis)?
- Gelden er eisen aan de koppeling tussen logische en fysieke testgevallen?
- Hoe wordt gegarandeerd dat testgevallen onafhankelijk van elkaar zijn (zie hoofdstuk 2) en hoe wordt omgegaan met uitzonderingen op het principe?
- Hoe belangrijk is onderhoudbaarheid van de testset?
- Welke hulpmiddelen zijn beschikbaar voor het administreren van de TestKubus?
- Hoe nauwkeurig moet het testtraject begroot kunnen worden?
- Moet de testset eenvoudig geautomatiseerd kunnen worden (dit vereist een eenduidige beschrijving van de in- en uitvoer van een testgeval)?
- Hoe groot wordt de testset?

Indien de opdrachtgever expliciete eisen aan de testset stelt, worden die requirements vanzelfsprekend opgenomen in het Mastertestplan (MTP). Het toepassen van de TestKubus en eventuele overige eisen die daaruit voortvloeien, hoeven niet per se in het MTP te worden genoemd: het is een aanpak om de testset te realiseren. Desondanks is afstemming van deze requirements met de opdrachtgever noodzakelijk: ook deze moet zich in het eindproduct kunnen vinden.

#### 4.1.2 Na de fixatie van het testplan

Een belangrijke activiteit tijdens de specificatie- en uitvoeringsfase is het bijhouden van ervaringscijfers voor de inspanning in uren voor de specificatie respectievelijk uitvoering per testgeval. Deze cijfers zijn essentieel voor de voortgangsbewaking, de begroting van het vervolg van het traject en de begroting van toekomstige releases.

Om extra inzicht te krijgen in de voortgang in de specificatiefase is het mogelijk om in het specificatieproces van een testgeval een fasering aan te brengen en per fase de relatieve inspanning (ervaringscijfers) aan te geven. Zie paragraaf 6.2 voor een voorbeeld van zo'n fasering.

Zowel in de specificatie- als in de uitvoeringsfase is het noodzakelijk dat de testteamleden hun voortgang nauwgezet bijhouden in de TestKubus, door voor elk gespecificeerd of uitgevoerd testgeval een status en/of testresultaat te registreren.

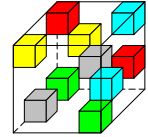
## 4.2 Fase Voorbereiding

De dataverzamelingen ten behoeve van de TestKubus worden aangelegd, bijvoorbeeld in de vorm van spreadsheets of het inrichten van een testspecificatietool. De gegevensstructuren worden neergezet en de lijsten met toegestane waarden voor de onderkende classificaties, releases, deelobjecten, testsoorten etc. worden ingevuld.

De TestKubus wordt toegankelijk gemaakt voor de testteamleden.

## 4.3 Fase Specificatie

De testgevallen worden gespecificeerd volgens de technieken zoals bepaald in de teststrategie. Voor elk testgeval wordt in de TestKubus een referentie opgenomen en alle classificaties worden ervoor gevuld (wanneer gebruik gemaakt wordt van een testspecificatietool is dit dikwijls geen speciale handeling maar onderdeel van het opvoeren van het testgeval).



Aan de hand van een detailintake testbasis kan de omvang van de testset worden begroot. Door zo snel mogelijk de testgevallen op logisch niveau te specificeren, zal de totale omvang van de testset duidelijk worden en kan de rest van het specificatietraject goed begroot en bewaakt worden. In dat geval kan tijdens het vervolg van de specificatiefase gebruik gemaakt worden van de zwaartecategorieën uit de TestKubus door in eerste instantie de testgevallen uit categorie 1 fysiek uit te werken.

Wanneer de omvang van de testset begroot is en de testteamleden hun voortgang adequaat bijhouden, biedt de TestKubus ogenblikkelijk inzicht in de stand van zaken, dat wil zeggen actuele informatie over het afgelegde en nog te gane traject.

Adequaat bijhouden van de voortgang maakt het voor de testmanager mogelijk om metrics op te bouwen om zodoende nog meer inzicht in en grip op het testtraject te krijgen.

#### 4.4 Fase Uitvoering

Voor de fase uitvoering levert de TestKubus de totale set aan uit te voeren testgevallen. In de fase Planning & Beheer is een inschatting gemaakt van de benodigde uitvoeringstijd per testgeval.

Evenals in de voorafgaande fase is het noodzakelijk dat de testteamleden de voortgang en de resultaten van de testuitvoering registreren in de TestKubus. De TestKubus levert dan direct inzicht in de voortgang, testdekking en/of risico's en maakt het mogelijk de gewenste metrics op te bouwen.

#### 4.5 Fase Afronding

In de fase Afronding wordt gerapporteerd over de kwaliteit van het testobject en testproces. Voor beide rapportages draagt de TestKubus de belangrijkste feitelijke informatie aan.

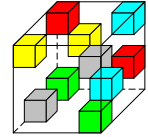
In de rapportages moeten tevens de ervaringscijfers voor de diverse kengetallen worden genoemd zoals die tijdens de specificatie- en uitvoeringsfase zijn opgesteld. Hoe beter de testmanager zijn kengetallen weet te onderbouwen, hoe inzichtelijker voor de organisatie de plannings- en begrotingen voor volgende testtrajecten zijn.

Bij de overdracht van de testware naar de beheerorganisatie moet extra aandacht besteed worden aan het overbrengen van de toegepaste aanpak voor het testwarebeheer: de TestKubus. De TestKubus zal pas goed gaan renderen in de beheersituatie en voorkomen moet worden dat het eenmaal goed ingeregelde instrument niet of onkundig wordt gebruikt.

Door tijdens de specificatiefase bij elk testgeval alle classificaties juist en volledig in te vullen ontstaat in de Testkubus stapsgewijs een gestructureerde testset. Het conserveren van testware neemt daardoor veel minder tijd in beslag en levert bovendien een hoger kwaliteitsniveau op dan in gangbare trajecten.

#### 4.6 Richtlijnen voor het gebruik van zwaartecategorieën

Het verdelen van testgevallen over zwaartecategorieën is geen exacte wetenschap en dit geldt ook voor de procentuele verdeling over de categorieën. Wel zijn er enkele richtlijnen om hier op een goede manier invulling aan te geven. Hierbij worden de volgende stappen onderkend:



1. Bepalen zwaartecategorieën,
2. Bepalen verdeelsleutel per zwaartecategorie,
3. Verdelen testgevallen over zwaartecategorieën.

Bij al deze stappen vormt steeds de situatie bij beheerreleases het referentiekader: het beoordelen van een reeds getest testobject waarin wijzigingen zijn aangebracht. De TestKubus is niet bedoeld om in nieuwbouwtrajecten te kunnen bepalen bij hoeveel testgevallen de applicatie 'een voldoende scoort'. Wanneer het in een nieuwbouwtraject om welke reden dan ook niet haalbaar blijkt om alle functionaliteit te testen, dan kan de TestKubus helpen om het aantal uit te voeren testgevallen beheerst terug te brengen. De risico's die dan genomen worden zijn echter van een geheel andere aard (vele malen groter) dan bij het met meer of minder diepgang beoordelen van een reeds getest testobject.

Alle stappen zijn gebaseerd op het beheersen van risico's (Risk Based (Regression) Testing): welke risico's wil men lopen en hoe wil men de risico's afdekken die men niet wil lopen? Het ligt daarom voor de hand deze stappen uit te laten voeren door de eigenaar van het systeem. Deze weet als geen ander, ongetwijfeld veel beter dan de tester, op welke niveaus hij zijn applicatie wil beoordelen, hoeveel testgevallen hij daarvoor nodig denkt te hebben en welke.

#### 4.6.1 Bepalen zwaartecategorieën

Om te beginnen moet bepaald worden op welke niveaus (toenemend in diepgang) het testobject in beheerreleases beoordeeld moet kunnen worden. Hierbij kan als leidraad dienen hoeveel gradaties van wijziging (met overeenkomstige risicoklassen) worden onderkend, bijvoorbeeld *ongewijzigd*, *meegewijzigd*, *klein*, *gemiddeld*, *groot*. Elk niveau waarop het testobject beoordeeld moet kunnen worden, wordt een zwaartecategorie, bijvoorbeeld: *pre-test*, *quick scan*, *kleine/gemiddelde/zware regressietest*, *volledige hertest*.

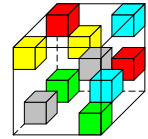
Het afleiden van de zwaartecategorieën van de onderkende gradaties van wijziging maakt het samenstellen van regressietestsets voor een beheerrelease heel doorzichtig. Voor de teststrategie wordt bepaald in welke mate een deelobject is gewijzigd, en daaruit volgt dan automatisch de bij die gradatie van wijziging behorende regressiezwaarte.

De zwaartecategorieën gelden voor alle deelobjecten in de testset. In het praktijkvoorbeeld uit hoofdstuk 2 waren dit er drie: *quick scan*, *regressietest* en *volledige hertest*. Een aantal van drie of vier zwaartecategorieën blijkt in de praktijk goed werkbaar.

#### 4.6.2 Bepalen verdeelsleutel per zwaartecategorie

Bij elke zwaartecategorie moet de vraag worden gesteld welk percentage van de testgevallen in de testset voldoende zou moeten zijn om de applicatie op dat niveau te kunnen beoordelen. Voorbeelden van deze vragen zijn:

- Welk deel van de testset moet uitgevoerd worden om te kunnen beoordelen of de applicatie kwalitatief voldoende is om verder te testen? (*pre-test*)
- Welk deel van de testset moet uitgevoerd worden om voor ongewijzigde deelobjecten te vast te stellen dat deze nog als vanouds werken? (*quick scan*)
- Welk deel van de testset moet uitgevoerd worden om voor licht gewijzigde of meegewijzigde deelobjecten te vast te stellen dat er geen regressie is opgetreden? (*gemiddelde regressietest*)



- Welk deel van de testset moet uitgevoerd op ingrijpend gewijzigde deelobjecten? (*volledige hertest*)

De antwoorden op deze vragen leveren een verdeelsleutel op, vaak in de vorm van een bandbreedte per categorie.

De op deze manier bepaalde bandbreedtes gelden voor alle deelobjecten in de testset. In het praktijkvoorbeeld uit hoofdstuk 2 waren dit: *quick scan* 10-15%, *regressietest* 60-70%, *volledige hertest* 100%.

#### 4.6.3 Verdelen testgevallen over zwaartecategorieën

Het verdelen van testgevallen over zwaartecategorieën gebeurt altijd per deelobject. Per zwaartecategorie, te beginnen met de laagste diepgang, wordt bepaald welke testgevallen uitgevoerd moeten worden om het deelobject op het gewenste niveau te kunnen beoordelen (elke volgende zwaartecategorie bevat ook de testgevallen van alle voorafgaande categorieën), bijvoorbeeld:

- Bij een *pre-test* of *quick scan*: enkele elementaire goedpaden.
- Bij een *lichte regressietest*: de testgevallen die de kernfunctionaliteit raken (bijvoorbeeld: doorlopen schermen, draaien batches, afdrukken gegevens, waarborgen referentiële integriteit, doorvoeren mutaties), en die niet al in een eerdere categorie zijn ingedeeld.

Is het aantal testgevallen dat langs deze weg voor deze categorie wordt geselecteerd groter dan het streefpercentage, dan moet worden uitgedund om te voorkomen dat de doorlooptijd te lang wordt. Dit kan veilig door te onderzoeken of er meerdere testgevallen in de testset zitten die hetzelfde testpad afdekken. Deze 'doublures' verhuizen naar de volgende zwaartecategorie.

Is het aantal testgevallen minder dan het streefpercentage, dan kan het zo worden gelaten maar de zwaartecategorie kan ook met extra testgevallen worden aangevuld tot het streefpercentage.

- Bij *zware regressietest*: eveneens de testgevallen gericht op de kernfunctionaliteit, maar met een uitgebreidere testdekking (bijvoorbeeld: doorlopen raadplegschermen, overige goed- en foutpaden, draaien eenmalige batches, afdrukken gegevens voor intern gebruik, testen gebruiksvriendelijkheid, testen performance), ook hier wordt geselecteerd uit testgevallen die niet al in een eerdere categorie zijn ingedeeld.

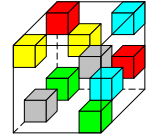
Ook hier kan worden uitgedund of aangevuld tot het streefpercentage op de manier zoals hiervoor beschreven.

- Bij een *volledige hertest*: alle nog niet ingedeelde testgevallen (bijvoorbeeld: theoretische of exotische testgevallen).

Op deze manier wordt voor elk deelobject een evenwichtige verdeling van testgevallen over zwaartecategorieën verkregen.

Wanneer aan een deelobject zo'n groot risico verbonden is, dat deze altijd met de grootste diepgang getest zal gaan worden, lijkt het segmenteren van de testset voor dat deelobject een administratieve opgave zonder toegevoegde waarde. Toch heeft het segmenteren ook in deze situatie zin omdat daarmee meer doeleinden gediend worden dan alleen het samenstellen van een (regressie)testset, bijvoorbeeld prioritering bij specificatie, uitvoering of automatisering, of het voorkomen van willekeurige uitdunning bij tijdgebrek. Bovendien kan bij een ongesegmenteerde testset niet nauwkeurig aangegeven worden wat het testen van het deelobject met





een lagere diepgang zou besparen. Juist in dergelijke onderhandelingen schuilt de kracht van de TestKubus.

#### 4.7 TestKubus bij een bestaande testset

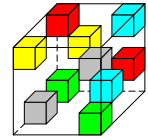
De voorafgaande paragrafen beschreven het van begin af aan opbouwen van een testset en het parallel daaraan inrichten van een TestKubus. Vaak zal de behoefte aan een gestructureerde testset pas ontstaan in een beheersituatie, op het moment dat de testset voor de applicatie al bestaat. Ook in deze situatie is het mogelijk om de TestKubus toe te passen.

Het volgende stappenscenario kan daarbij gehanteerd worden:

1. De haalbaarheid van de TestKubus moet worden bepaald. Dit is afhankelijk van verschillende factoren, zie paragraaf 4.1.1, maar wordt bovendien beïnvloed door de bevindingen uit het volgende punt.
2. De bestaande testset moet geëvalueerd worden:
  - Is de testset nog altijd een goede afspiegeling van de applicatie? Indien dit niet (meer) het geval is, moet inspanning getroost worden om het te herstellen. Dit is een belangrijke randvoorwaarde, zie paragraaf 3.1.
  - Is de testset strijdig met het onafhankelijkheidsprincipe van de TestKubus? Zo ja, dan moeten testgevallen herschreven worden. Uitgangspunt is geen afhankelijkheden tussen testgevallen.
3. De requirements voor de TestKubus moeten worden opgesteld.
4. De dataverzamelingen voor de TestKubus moeten worden aangelegd en de classificaties voor elk testgeval aangebracht.

Deze laatste stap kan tamelijk bewerkelijk zijn indien er veel classificaties worden onderkend. Het best kan begonnen worden met twee classificaties: zwaartecategorie en deelobject.

Wanneer alleen een ongestructureerde of onvolledige testset beschikbaar is, is de situatie niet hopeloos. Het inrichten van een TestKubus helpt om de ontbrekende structuur aan te brengen. Later, wanneer de testset in een TestKubus is ondergebracht, zal snel duidelijk zijn voor welke deelobjecten te weinig of juist teveel testgevallen zijn opgenomen. Dit vormt dan een goede aanleiding om de testset in overeenstemming met de teststrategie te brengen, of, als die er nog niet is, om een teststrategie op te gaan stellen.



## 5. HET GEBRUIK VAN DE TESTKUBUS IN ONDERHOUD

In dit hoofdstuk wordt het gebruik van de TestKubus in beheer beschreven. Net als in het vorige hoofdstuk zijn de stappen opnieuw gerelateerd aan de TMap-fasering.

### 5.1 Fase Planning & Beheer

Voor elke onderhoudsrelease moet de omvang en de impact van de wijzigingen op de bestaande applicatie bepaald worden. Aan de hand daarvan moet een teststrategie voor de onderhoudsrelease worden opgesteld.

Er moet geverifieerd worden of de testset in de TestKubus nog actueel is en aansluit op de gekozen teststrategie (zie hierover paragraaf 3.1). Voor de nieuwe en gewijzigde functionaliteit moeten aanvullende testgevallen worden opgesteld.

De TestKubus biedt de mogelijkheid om op de verhoging van de foutkans in een deelobject te reageren door per deelobject de zwaartecategorie voor de test vast te stellen, bv.:

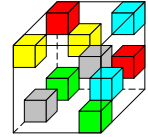
- Ongewijzigde deelobjecten worden getest met zwaarte 1 (alleen de testgevallen met zwaartecategorie 1 worden uitgevoerd).
- Licht gewijzigde deelobjecten worden getest met zwaarte 2 (alleen de testgevallen met zwaartecategorie 1 en 2 worden uitgevoerd).
- Ingrijpend gewijzigde deelobjecten worden getest met zwaarte 3 (alle testgevallen worden uitgevoerd).

Op basis van de zwaarte van de afzonderlijke deelobjecten kan in de TestKubus de exacte omvang van de testset worden bepaald bij een gegeven teststrategie. Dit in combinatie met een begroting van het aantal nieuwe testgevallen voor de nieuwe of gewijzigde functionaliteit is de omvang van de nieuwe release. Met behulp van de kengetallen voor testspecificatie en –uitvoering kan nu een nauwkeurige begroting van het testtraject worden gemaakt. Deze begroting is zo betrouwbaar als de kengetallen die zijn bepaald tijdens eerdere releases.

Indien de opdrachtgever beperktere resources beschikbaar stelt, kan met behulp van de TestKubus gemakkelijk worden bepaald wat de besparing is bij het verlagen van de zwaarte van het deelobject. De opdrachtgever krijgt hiermee al in de planningsfase een concreet beeld van de risico's die hij neemt en de kosten die met het afdekken ervan gemoeid zijn. Eventueel kan de zwaarte *optioneel* verlaagd worden. Wanneer dan later bij de testuitvoering significant meer bevindingen gedaan worden dan in andere deelsystemen, gaat er alsnog een gefundeerde aanvraag voor extra resources uit om het deelobject met een hogere zwaarte te testen.

De uiteindelijke invulling van de testset (welke deelobjecten met welke zwaarte) wordt vastgelegd in een testplan.

Indien metrics beschikbaar zijn m.b.t. het specificeren, aanpassen en uitvoeren van testgevallen, is het mogelijk het testtraject nauwkeurig in te plannen. Toepassing van onafhankelijkheidsprincipe (zie hoofdstuk 2) vereenvoudigt het inplannen van de testuitvoeringsfase: er hoeft dan immers met weinig of geen onderlinge verbanden tussen testgevallen rekening gehouden te worden.



## 5.2 Fase Voorbereiding

De dataverzamelingen ten behoeve van de TestKubus worden voorbereid op de aankomende onderhoudsrelease.

## 5.3 Fase Specificatie

Het specificeren van de testgevallen voor de nieuwe of gewijzigde functionaliteit gebeurt niet anders dan bij het inrichten van de TestKubus, dus conform de gekozen teststrategie (zie paragraaf 4.3).

Testgevallen voor functionaliteit die is komen te vervallen, wordt geschoond (bv. door ze de status *vervallen* toe te kennen, waarbij de testuitvoeringshistorie behouden blijft).

De testset voor de onderhoudsrelease wordt samengesteld door per deelobject een subset van de bestaande testgevallen te selecteren in overeenstemming met de afgesproken zwaarte.

## 5.4 Fase Uitvoering

In eerste instantie worden de testgevallen voor de nieuwe en gewijzigde functionaliteit uitgevoerd.

Voor het overige gebeurt het uitvoeren van de testgevallen niet anders dan bij het inrichten van de TestKubus, zie paragraaf 4.4.

## 5.5 Fase Afronding

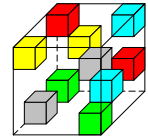
Wanneer het testen van de onderhoudsrelease is afgerond, maken de testgevallen voor nieuwe en gewijzigde functionaliteit automatisch deel uit van de geconsolideerde testset voor de applicatie. Voor elk deelobject dat in de onderhoudsrelease aangepast is, moet de testset opnieuw gevalideerd worden:

- Alle testgevallen voor een deelobject zijn opgeslagen in dezelfde directory (met correcte verwijzingen daarnaartoe in de TestKubus) of repository-node.
- De verdeling van testgevallen over zwaartecategorieën is in overeenstemming met de gestelde normen.

Er wordt geverifieerd of de gebruikte kengetallen voor testspecificatie en testuitvoering bijstelling behoeven.

Voor het overige zie de afronding bij het inrichten van de TestKubus, paragraaf 4.5.

Door toepassing van de TestKubus evolueert de testset in de beheersituatie automatisch mee met het testobject.



## 6. RAPPORTAGE & METRICS

Een testtraject is voor buitenstaanders vaak een onduidelijk en ongrijpbaar traject. Er zijn dan ook veel vragen waar de testmanager antwoord op moet geven. Om deze vragen goed te kunnen beantwoorden is het noodzakelijk om ze te onderbouwen met feiten. Actuele en gefundeerde ervaringscijfers (metrics) bieden hier uitkomst.

Metrics dienen niet alleen plannings- en begrotingsdoeleinden, maar kunnen tevens dienen als graadmeter om testtrajecten te spiegelen aan de normen van de organisatie

In nieuwbouw-testtrajecten worden metrics tijdens de looptijd van het traject opgebouwd. Omdat het bij metrics om de grote getallen gaat, beschikt men pas na verloop van enige tijd over betrouwbare ervaringscijfers. In de aanloopfase blijft de toegevoegde waarde van metrics beperkt.

Bij releasematig testen kunnen metrics eenvoudig opgebouwd worden op basis van cijfers uit vorige releases. Hiermee is het mogelijk om metrics op te bouwen voor zowel de kwaliteit van het testobject als voor de planning van het testtraject (zie TMap, paragraaf 23.5-6). Belangrijke metrics in de TestKubus zijn:

- Benodigde tijd voor het maken van een nieuw testgeval
- Benodigde tijd voor het wijzigen van een testgeval
- Benodigde tijd voor het uitvoeren van een testgeval
- Foutersteltijd

### 6.1 Plannen en begroten

Wanneer een onderhoudsrelease gedefinieerd is, moet de impact bepaald worden op testobject en testset. Het moeilijkste hierbij is het bepalen van de omvang van de testrelease. Door de gestructureerde opbouw van de testset kan met behulp van de bij de TestKubus bijgehouden ervaringscijfers vrij eenvoudig bepaald worden wat de hoeveelheid testwerk voor een bepaalde onderhoudsrelease zal zijn.

Wanneer men weet wat de omvang van het uit te voeren werk is, kan men met de verzamelde metrics bepalen wat de doorlooptijd en de daaraan gerelateerde kosten zullen zijn. De nauwkeurigheid van de planning staat of valt echter bij het goed inschatten van de uit te voeren werkzaamheden.

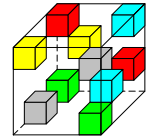
### 6.2 Rapportage

De testmanager heeft gedurende het testtraject constant behoefte om geïnformeerd te zijn over de voortgang. Niet alleen om te kunnen rapporteren naar de opdrachtgever, maar vooral om gevoel te houden bij het testtraject en om vroegtijdig eventuele noodzakelijke maatregelen te kunnen nemen.

Er kan worden gerapporteerd op de eindresultaten van de verschillende activiteiten (zoveel testgevallen gespecificeerd, zoveel testgevallen uitgevoerd met resultaat OK). Omvangrijke activiteiten, zoals testspecificatie, kunnen door middel van een Work Breakdown Structure (WBS) worden onderverdeeld in kleinere. De detaillering van de rapportage kan worden verfijnd naar de detaillering van deze WBS. Een WBS kan worden opgenomen in de TestKubus en door de testers worden bijgehouden bij het vastleggen van hun voortgang.

TestKubus

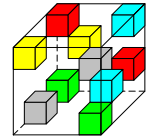
Rapportage & metrics



Een praktijkvoorbeeld van een WBS voor de fase testspecificatie:

- 10% identificeren testgeval
- 10% specificeren logisch testgeval
- 20% specificeren testdata
- 50% opstellen testscript
- 10% bijwerken TestKubus

Met behulp van een dergelijke WBS kunnen gedetailleerdere voortgangsrapportages gemaakt worden.



## 7. HULPMIDDELEN

In concept is de TestKubus een verzameling aanvullende gegevens bij testgevallen (zie hoofdstuk 2). Het vastleggen van deze gegevens kan op verschillende manieren met verschillende ondersteunende tools.

### 7.1 Spreadsheets

Het goedkoopste en tegelijk meest praktische tool is een spreadsheet (bijvoorbeeld MS Excel). Spreadsheets zijn standaard op de meeste werkplekken geïnstalleerd, gemakkelijk in het gebruik en bevatten krachtige rekenfuncties waarmee snelle en nauwkeurige rapportages mogelijk zijn.

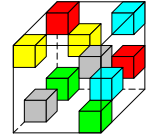
Een eenvoudige TestKubus zou er als volgt uit kunnen zien (uitgebreidere voorbeelden zijn beschikbaar bij de leden van de expertisegroep TestKubus):

release 1.1	release 1.2	release 1.3	Vervallen	Identificatie	TG Naam	Zwaartecategorie 1	Zwaartecategorie 2	Zwaartecategorie 3	Deelobject	Testdatum	Tester	Resultaat	Bevindingen
x	x	x		1	Invoer naam	x	x	x	Invoer	23-01-05	PWN	OK	
	x			2	Invoer naam met leestekens			x	Invoer				
	x	x		3	leeglaten naam		x	x	Invoer	23-01-05	PWN	NOK	437
x	x	x		4	tekstfile invoer	x	x	x	Interfaces	05-02-05	HDT	OK	
	x	x		5	spreadsheet invoer		x	x	Interfaces	05-02-05	HDT	OK	
			x	6	etc.								

- In het bovenstaande voorbeeld zijn de zwaartecategorieën van de TestKubus als aparte kolommen opgenomen, en niet als verschillende waarden in één kolom. Dit is gedaan om het selecteren van de testgevallen te vergemakkelijken.
- Uit het voorbeeld blijkt dat deelobject *Invoer* in rel. 1.1 getest is met zwaarte 1, in rel. 1.2 met zwaarte 3 en in release 1.3 met zwaarte 2. De gegevens met betrekking tot de testuitvoering slaan op de huidige release. Bij overgang naar een volgende release worden ze gearchiveerd.

Bij het gebruik van spreadsheets moet bewaakt worden dat de administratie van de testgevallen in de TestKubus (het spreadsheet) synchroon blijft lopen met de feitelijke specificaties in bijvoorbeeld tekstdocumenten. Een ander risico is dat de werkbaarheid van spreadsheets afneemt met de omvang van de testset en berekeningen daarop.

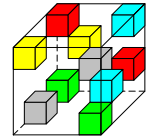
Op dit laatste punt zijn databasesystemen (bv. MS Access) in het voordeel, maar het nadeel van een databasesysteem is dat er eerst het nodige geprogrammeerd moet worden en dat data-invoer moeizamer verloopt dan bij spreadsheets. Ook zijn dergelijke tools vaak niet standaard geïnstalleerd en wordt het ongecontroleerd gebruik ervan in veel organisaties ontmoedigd.



## 7.2 Testspecificatietools

Wanneer in een organisatie gebruik gemaakt wordt van een testspecificatietool (bijvoorbeeld TestDirector), kan de TestKubus in hetzelfde tool worden ondergebracht. Voorwaarde is dat het tool de mogelijkheid kent om zelfgedefinieerde velden aan testgevallen toe te voegen (nodig voor de classificaties van de TestKubus). Vaak bezitten deze tools standaard rapportagefuncties die ook voor de TestKubus gebruikt kunnen worden.

Het voordeel van een TestKubus geïntegreerd in een testspecificatietool is dat het administreren van de aanvullende kenmerken en de voortgang in veel gevallen automatisch verloopt.



## 8. DE TESTKUBUS IN DE PRAKTIJK

De TestKubus is ontwikkeld en voor het eerst toegepast bij een van de klanten van Sogeti Nederland B.V. De volgende paragrafen bevatten een verslag van het gebruik van de TestKubus in dit praktijkgeval.

### 8.1 Uitgangssituatie

#### Klantorganisatie

De betreffende organisatie is de grootste speler in zijn branche in Nederland. Ze begon echter steeds meer concurrentie te ondervinden. Om deze concurrentie voor te blijven, was het noodzakelijk om snel nieuwe functionaliteit te kunnen implementeren. Naast deze 'time-to-market'-uitdaging had de klant problemen met de kwaliteit van de opgeleverde systemen.

#### Te testen systeem

Het te testen systeem was de core-applicatie van de klant. De informatie in het systeem was het kapitaal van het bedrijf. Het was een complex systeem met interfaces naar verschillende externe partijen met uiteenlopende eisen en wensen.

Het ging om een pakkettoepassing die door een buitenlandse softwareleverancier werd ontwikkeld. De klant wilde snel nieuwe oplossingen die echter niet in de standaard pakketfunctionaliteit zaten, wat leidde tot steeds meer maatwerk. De leverancier had moeite dit maatwerk snel en met voldoende kwaliteit te leveren.

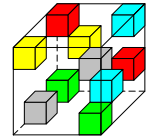
Het systeemontwikkelingsproces schoot ook aan de kant van de klant tekort. Het requirementsproces was niet goed onder controle. Functionele wijzigingen werden laat in het ontwikkeltraject doorgevoerd en documentatie ervan liet te wensen over. Kennis van de functionaliteit van het systeem zat voornamelijk in hoofden van mensen.

Het testproces kenmerkte zich door lange doorlooptijden (vijf weken voor een regressietest), was onvoldoende beheersbaar en had een slecht onderhoudbare, verouderde testset.

Voor de nieuwe releases was allereerst een nieuwe testset nodig. Door deze testset als deliverable te benaderen, werd de klant gemotiveerd om eisen op te stellen waaraan de nieuwe testset moest voldoen:

1. De testset moet sneller doorlopen kunnen worden met een verantwoorde dekking.
2. De testset moet door testers zonder materiekennis uitgevoerd kunnen worden.
3. De testset moet flexibel zijn.
4. Het moet mogelijk zijn om te testen op basis van de in een release onderkende risico's.
5. De testset moet op termijn gemakkelijk omgezet kunnen worden naar een geautomatiseerde testset.
6. De testset moet overdraagbaar en onderhoudbaar zijn.





## 8.2 Oplossingen

Om tegemoet te komen aan de genoemde eisen, werden de volgende oplossingen uitgedacht:

1. De testset werd modulair op testgevalniveau. Het principe van modulariteit houdt in dat testgevallen onafhankelijk van elkaar uit te voeren zijn (eis 1 en 5).
2. De testgevallen werden op een zodanig niveau van detaillering uitgewerkt dat een persoon zonder materiekennis ze kon uitvoeren en ze relatief eenvoudig geautomatiseerd konden worden (eis 2 en 5).
3. Het aantal testacties per testgeval werd geminimaliseerd (eis 1 en 3).
4. De testgevallen werden geassocieerd (eis 1, 3 en 4).
5. Er werd een administratief systeem opgezet waarmee de testset te onderhouden was. Dit administratieve systeem bevatte tevens een repository waarmee herleidbaar was welk testgeval welke functionaliteit afdekte (eis 4 en 6).

Door het implementeren van deze oplossingen en de verdere verfijning ervan ontstond een geheel nieuwe aanpak voor het beheren van testsets: de TestKubus.

## 8.3 Resultaten

Door toepassing van de TestKubus-aanpak zijn bij deze klant de volgende voordelen behaald:

- De doorlooptijd van testspecificatie en –uitvoering is sterk verkort (80% korter dan voorheen) door:
  - modulair opgebouwde testsets (parallele testuitvoering)
  - hergebruik van testscripts
  - op basis van risico's samengestelde regressietestsets (verantwoorde dekking met veel minder testgevallen)
  - efficiënte inzet van extra testresources (de TestKubus maakt inzichtelijk in welke mate een extra testresource bijdraagt aan een verkorting van de doorlooptijd).

Testen was niet langer het struikelblok voor een snelle time-to-market.

- Testuitvoering is goedkoper geworden door de inzet van testers zonder materiekennis. Er zijn minder problemen bij het invullen van de testresources.
- De effecten op kosten en kwaliteit van tussentijdse wijzigingen van de teststrategie zijn ogenblikkelijk inzichtelijk.
- De aansturing van de testuitvoering is eenvoudiger geworden door het nagenoeg ontbreken van afhankelijkheden tussen testgevallen.
- De voortgang van testspecificatie en –uitvoering is op elk willekeurig ogenblik in het testtraject transparant.
- Adequate, op businessrisico's gestuurde rapportages richting opdrachtgever en andere belanghebbenden zijn beschikbaar gekomen.
- Er zijn metrics (ervaringscijfers, kengetallen) beschikbaar gekomen waarmee testtrajecten nauwkeurig begroot en gepland kunnen worden (tot op een halve dag nauwkeurig).

Het grootste compliment voor de TestKubus kwam van de kant van de opdrachtgever, een manager van de klantorganisatie. Voor hem was testen nu niet langer de meest ongreepbare en ondoorzichtige activiteit in de IT-organisatie: dankzij de TestKubus was testen voor hem de enige *echt* transparante projectactiviteit geworden.