



Chris C. Schotanus
TestFrame, een methode voor gestructureerd testen
Voorjaarsevent Testnet: 22 juni 2009

Samenvatting:

Meer dan 12 jaar geleden is Logica begonnen met het ontwikkelen van een testmethode waarmee organisaties op een gestructureerde manier hun testen konden ontwikkelen en uitvoeren: TestFrame®. In 1999 heeft dit geresulteerd in een eerste boek waarin de TestFrame testmethode werd beschreven.

Ondertussen is er veel in de IT markt gebeurd. Nieuwe trends in ontwikkelmethoden dienden zich aan, denk bijvoorbeeld aan Service Oriented Architectures (SOA). Ook het testvak is meer volwassen geworden. Zo is testen geëvolueerd van een activiteit die een ontwikkelaar 'erbij' deed tot een volwaardige discipline die ondersteunt bij het valideren en verifiëren van de opgeleverde producten aan de eindgebruiker. Er bestaat nu zelfs een wereldwijd gestandaardiseerde testcertificering: ISTQB.

Al deze ontwikkelingen hebben Logica ertoe gebracht de testmethode TestFrame te vernieuwen. Het boek dat u in handen heeft, is vooral geschreven voor de praktijk. Dit is een verschil met de versie uit 1999 waarin vooral de principes van de methode werden uitgewerkt. Alle stappen binnen deze gestructureerde testmethode komen uitgebreid aan de orde. U zult merken dat het boek u niet alleen helpt uw effectiviteit te verhogen, maar u ook door haar sterke focus op hergebruik een handvat biedt om meer efficiënt te werken. Hierdoor blijft TestFrame dé praktische handleiding voor het testen van informatiesystemen. In deze presentie wordt ingegaan op de verschillen ten opzichte van de vorige versie van TestFrame.

Biografie

Chris C. Schotanus heeft meer dan dertig jaar ervaring in de IT, waarvan vijftien op het gebied van testen, testmanagement, inrichten van testorganisaties en testautomatisering. Als senior management consultant adviseert Chris organisaties in de financiële sector, de overheid, telecommunicatie, binnen Nederland en daarbuiten op het gebied van testbeleid en testorganisatie. Hij is medeontwikkelaar en coauteur van Logica's aanpak op dit gebied: TestGrip. Grip op kwaliteit en processen in IT door testbeleid en testorganisatie (Marselis et al., 2007). Tevens is hij een van de grondleggers van Logica's aanpak voor gestructureerd testen,

TestFrame®, waarvan dit boek het resultaat is. Zijn bijdragen zijn ook terug te vinden in Logica's boek Succesvol testmanagement: een integrale aanpak (Van de Burgt, 2003) en hij heeft een hoofdstuk geschreven voor The Testing Practitioner (Van Veenendaal, 2002). Chris is docent van verschillende testgerelateerde cursussen en verzorgt regelmatig presentaties op verschillende (inter)nationale congressen.

Releasing your potential
logica



TestFrame

Chris C. Schotanus
Senior Management Consultant
Testing & Test management
chris.schotanus@logica.com

2-6-2009 TestFrame © Logica 2009. All rights reserved. 1

logica

Waar waren we ook alweer gebleven

1995



- Aanpak wordt ontwikkeld bij AEX en EOE
- Begonnen als werkwijze voor geautomatiseerd testen
- Verbetering t.o.v. standaard C&P tools

2-6-2009 TestFrame 3

logica

Een stukje historie

1995

1999

2008





2-6-2009 TestFrame 2

logica

Waar waren we ook alweer gebleven

1999

- Naam gewijzigd in TestFrame
- Boek gepubliceerd
- Meer gericht op methode
- Inclusief test management



2-6-2009 TestFrame 4

logica

En waar gaan we verder

2008

- Nieuw boek
- Minder over automatisering
- Geen test management
- Relatie met RRBT
- Meer praktische voorbeelden



2-6-2009 TestFrame 5

logica

En waar gaan we verder

2009

- Nieuw boek
- Minder over automatisering
- Geen test management
- Relatie met RRBT
- Meer praktische voorbeelden
- En de Engelse versie

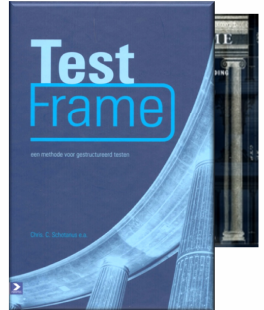


2-6-2009 TestFrame 5

logica

Belangrijkste wijzigingen tov 1999

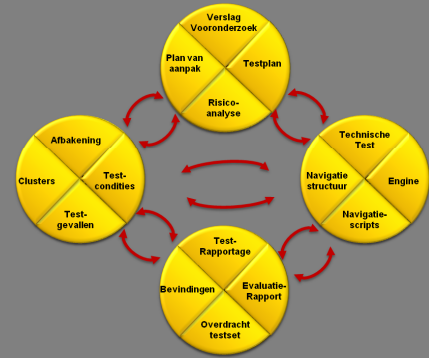
- Maar praktische beschrijving van het "hoe"
 - Veel voorbeelden
 - Praktijkgericht
- Nadruk op methode ipv testautomatisering
 - Geen beschrijving van navigatie
 - Wel handmatige testuitvoering
- Verbetering aansluiting bij RRBT
 - Overerving risico's en prioriteiten
 - Rapportage afgestemd op teststrategie
 - Projectie TF faseringsmodel op RRBT model



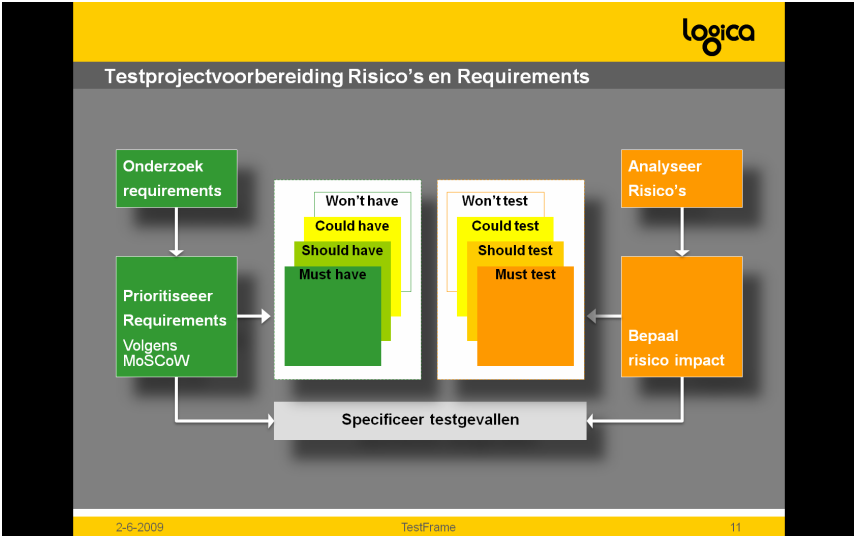
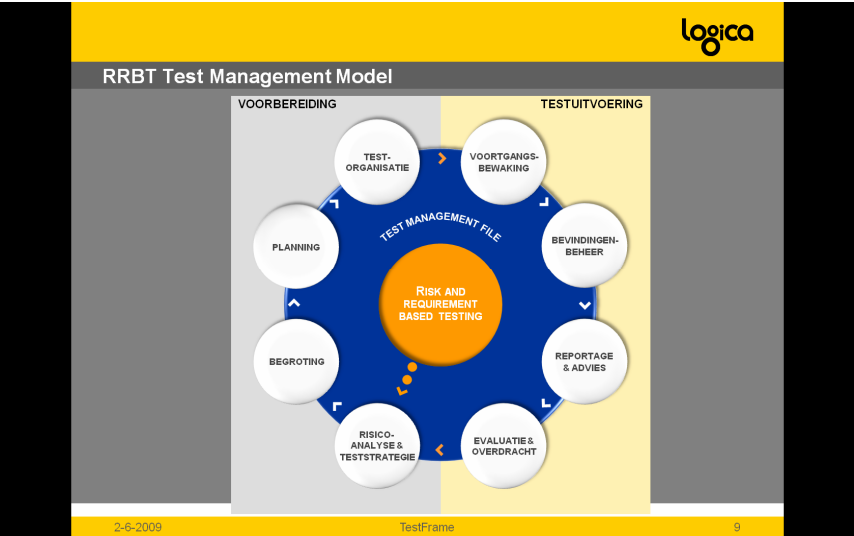
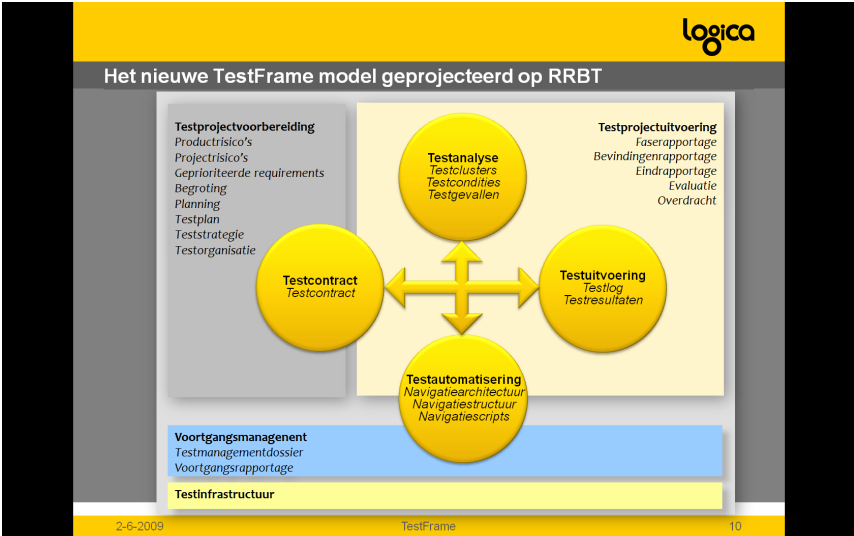
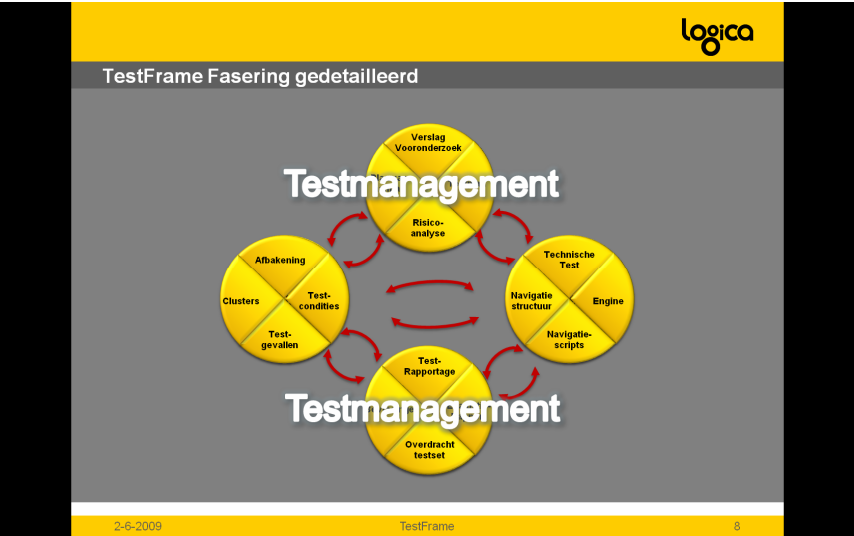
2-6-2009 TestFrame 6

logica

TestFrame Fasering gedetailleerd



2-6-2009 TestFrame 8



Prioriteit van testen

Een requirement krijgt twee prioriteiten:

- Gewenstheid
- Productrisico

Een testconditie “erft” de risicoprioriteit.

TestFrame is een requirement gebaseerde aanpak!

Testcondities erven de prioriteiten van de risico's die tijdens de PRA zijn gedefinieerd.

Rapportage gebeurt o.b.v. deze risico's

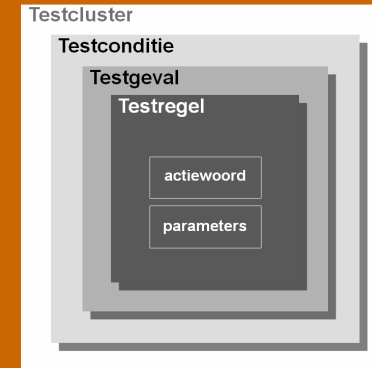
Clustermatrix: Stakeholder per Testsoort

Stakeholder	Eind-gebruiker	Support	Beheer	IT afdeling	Interne controle
Testsoort					
GAT/PAT	Prakikbaarheid Leerbaarheid		Tijdsbeslag Inpasbaarheid		
SIT					Nauwkeurigheid betrouwbaarheid
ST	Nauwkeurigheid geschiktheid	Analyseerbaarheid		Herstelbaarheid Wijzigbaarheid	
CT					

Clusterkaarten per matrixkruispunt

	Aspect informatie	De informatie wordt verzameld uit verschillende bronnen.
	Doelstelling	Omgekeerde aanpak met feedback.
	Doelgroep	De informatie wordt aan bestuurders overgedragen.
	Testplan	De testplan wordt aan bestuurders overgedragen.
	Testresultaten	De testplan wordt aan bestuurders overgedragen.
	Opdracht	De opdracht wordt aan bestuurders overgedragen.
	Testplan	De testplan wordt aan bestuurders overgedragen.
Beoordeling	Testplan	De testplan wordt aan bestuurders overgedragen.
	Testresultaten	De testplan wordt aan bestuurders overgedragen.
Resultaat	Testplan	De testplan wordt aan bestuurders overgedragen.
	Testresultaten	De testplan wordt aan bestuurders overgedragen.
Uitgangsmateriaal	Testplan	De testplan wordt aan bestuurders overgedragen.
	Testresultaten	De testplan wordt aan bestuurders overgedragen.
Uitvoering	Testplan	De testplan wordt aan bestuurders overgedragen.
	Testresultaten	De testplan wordt aan bestuurders overgedragen.
Testomgeving	Testplan	De testplan wordt aan bestuurders overgedragen.
	Testresultaten	De testplan wordt aan bestuurders overgedragen.
Resultaat	Testplan	De testplan wordt aan bestuurders overgedragen.
	Testresultaten	De testplan wordt aan bestuurders overgedragen.

Analyse - hierarchie



STAP 1

BRD1	AFDELING BROOD
GRN1	AFDELING GROENTE

STAP 2

BDRD1 – Afdeling Brood		PRIO
BRD1C01	Het moet mogelijk zijn brood in te voeren	Must
BRD1C02	Het moet mogelijk zijn gebak in te voeren	Must
BRD1C03	Het moet mogelijk zijn om brood te wijzigen	Must

STAP 3

BDRD1 – Afdeling Brood			
BRD1C02 - Het moet mogelijk zijn gebak in te voeren			
BRD1C02T01	Kwarktaar wordt ingevoerd		
	Soort	Aantal	Prijs
Voer gebak in	Kwarktaart	4	3,00

Testcluster

Testconditie

Testgeval

Test-regel



Handmatige testuitvoering

Cluster	BRD1 AFDELING BROOD		
Versie	0.1		
Datum	1/12/2005		
Auteur	J. Jansen		
start applicatie			
testconditie	BDR1C01	Het moet mogelijk zijn	
testgeval	BRD1C01T01	Invoeren van brood	
	Soort	Aantal	Prijs
Voer brood in	bruin	2	2.50
	Soort	Aantal	Prijs
Controleer brood	bruin	2	2.50
testcase	C1C1T02	Invoeren van volkoren	
	Soort	Aantal	Prijs
Voer brood in	volkoren	2	3.50

Actiewoorddocumentatie

Actiewoord	Voer brood in
Omschrijving	Voer de eigenschappen van brood in
Pre conditie	Geen subwindows open
Startscherm	hoofdscherm

ACTIES

klik op [Nieuw product]
op [producten] selecteer [brood]

Soort = **Arg-2** bruin
Aantal = **Arg-3** 2
Prijs = **Arg-4** 2.50

klik op knop [ok]
klik op [ok] in [bevestiging]
klik op [end]

Eindscherm	hoofdscherm
Postconditie	Alle subwindows gesloten

2-6-2009

TestFrame

18

The diagram illustrates the TestFrame architecture, which automates test execution. It consists of several key components and their interactions:

- System under test:** The component being tested, represented by a yellow box at the bottom left.
- TestFrame Engine:** The central component, represented by a yellow diamond, which manages the test execution process.
- TestFrame Report of Execution of Test:** A document generated by the engine, showing test results. It includes a header with the TestFrame logo and a table of test results.
- TestFrame GUI:** A graphical user interface for managing test cases, represented by a yellow box at the bottom right.

The flow of data is as follows:

- The **System under test** provides input to the **TestFrame Engine**.
- The **TestFrame Engine** generates a **TestFrame Report of Execution of Test**.
- The **TestFrame Engine** also interacts with the **TestFrame GUI** for test case management.

TestFrame Report of Execution of Test:

Testcase	Result	Message
Testcase 1	Pass	Testcase 1 passed
Testcase 2	Fail	Testcase 2 failed
Testcase 3	Pass	Testcase 3 passed
Testcase 4	Fail	Testcase 4 failed
Testcase 5	Pass	Testcase 5 passed
Testcase 6	Fail	Testcase 6 failed
Testcase 7	Pass	Testcase 7 passed
Testcase 8	Fail	Testcase 8 failed
Testcase 9	Pass	Testcase 9 passed
Testcase 10	Fail	Testcase 10 failed
Testcase 11	Pass	Testcase 11 passed
Testcase 12	Fail	Testcase 12 failed
Testcase 13	Pass	Testcase 13 passed
Testcase 14	Fail	Testcase 14 failed
Testcase 15	Pass	Testcase 15 passed
Testcase 16	Fail	Testcase 16 failed
Testcase 17	Pass	Testcase 17 passed
Testcase 18	Fail	Testcase 18 failed
Testcase 19	Pass	Testcase 19 passed
Testcase 20	Fail	Testcase 20 failed
Testcase 21	Pass	Testcase 21 passed
Testcase 22	Fail	Testcase 22 failed
Testcase 23	Pass	Testcase 23 passed
Testcase 24	Fail	Testcase 24 failed
Testcase 25	Pass	Testcase 25 passed
Testcase 26	Fail	Testcase 26 failed
Testcase 27	Pass	Testcase 27 passed
Testcase 28	Fail	Testcase 28 failed
Testcase 29	Pass	Testcase 29 passed
Testcase 30	Fail	Testcase 30 failed
Testcase 31	Pass	Testcase 31 passed
Testcase 32	Fail	Testcase 32 failed
Testcase 33	Pass	Testcase 33 passed
Testcase 34	Fail	Testcase 34 failed
Testcase 35	Pass	Testcase 35 passed
Testcase 36	Fail	Testcase 36 failed
Testcase 37	Pass	Testcase 37 passed
Testcase 38	Fail	Testcase 38 failed
Testcase 39	Pass	Testcase 39 passed
Testcase 40	Fail	Testcase 40 failed
Testcase 41	Pass	Testcase 41 passed
Testcase 42	Fail	Testcase 42 failed
Testcase 43	Pass	Testcase 43 passed
Testcase 44	Fail	Testcase 44 failed
Testcase 45	Pass	Testcase 45 passed
Testcase 46	Fail	Testcase 46 failed
Testcase 47	Pass	Testcase 47 passed
Testcase 48	Fail	Testcase 48 failed
Testcase 49	Pass	Testcase 49 passed
Testcase 50	Fail	Testcase 50 failed
Testcase 51	Pass	Testcase 51 passed
Testcase 52	Fail	Testcase 52 failed
Testcase 53	Pass	Testcase 53 passed
Testcase 54	Fail	Testcase 54 failed
Testcase 55	Pass	Testcase 55 passed
Testcase 56	Fail	Testcase 56 failed
Testcase 57	Pass	Testcase 57 passed
Testcase 58	Fail	Testcase 58 failed
Testcase 59	Pass	Testcase 59 passed
Testcase 60	Fail	Testcase 60 failed
Testcase 61	Pass	Testcase 61 passed
Testcase 62	Fail	Testcase 62 failed
Testcase 63	Pass	Testcase 63 passed
Testcase 64	Fail	Testcase 64 failed
Testcase 65	Pass	Testcase 65 passed
Testcase 66	Fail	Testcase 66 failed
Testcase 67	Pass	Testcase 67 passed
Testcase 68	Fail	Testcase 68 failed
Testcase 69	Pass	Testcase 69 passed
Testcase 70	Fail	Testcase 70 failed
Testcase 71	Pass	Testcase 71 passed
Testcase 72	Fail	Testcase 72 failed
Testcase 73	Pass	Testcase 73 passed
Testcase 74	Fail	Testcase 74 failed
Testcase 75	Pass	Testcase 75 passed
Testcase 76	Fail	Testcase 76 failed
Testcase 77	Pass	Testcase 77 passed
Testcase 78	Fail	Testcase 78 failed
Testcase 79	Pass	Testcase 79 passed
Testcase 80	Fail	Testcase 80 failed
Testcase 81	Pass	Testcase 81 passed
Testcase 82	Fail	Testcase 82 failed
Testcase 83	Pass	Testcase 83 passed
Testcase 84	Fail	Testcase 84 failed
Testcase 85	Pass	Testcase 85 passed
Testcase 86	Fail	Testcase 86 failed
Testcase 87	Pass	Testcase 87 passed
Testcase 88	Fail	Testcase 88 failed
Testcase 89	Pass	Testcase 89 passed
Testcase 90	Fail	Testcase 90 failed
Testcase 91	Pass	Testcase 91 passed
Testcase 92	Fail	Testcase 92 failed
Testcase 93	Pass	Testcase 93 passed
Testcase 94	Fail	Testcase 94 failed
Testcase 95	Pass	Testcase 95 passed
Testcase 96	Fail	Testcase 96 failed
Testcase 97	Pass	Testcase 97 passed
Testcase 98	Fail	Testcase 98 failed
Testcase 99	Pass	Testcase 99 passed
Testcase 100	Fail	Testcase 100 failed

TestFrame

The diagram illustrates the relationship between RBT (Requirement-Based Testing) and TestFrame. On the left, a circular process for RBT is shown, centered around 'RBT AND REQUIREMENT-BASED TESTING'. The process is divided into two main phases: 'VOORBEREIDING' (Preparation) and 'TESTUITVOERING' (Test Execution). The steps are: Plan, Prepare, Execute, Evaluate, and Report. On the right, the TestFrame diagram is shown, consisting of four main components: Test contract, Test analysis, Test execution, and Test automation, all interconnected by a central double-headed arrow.

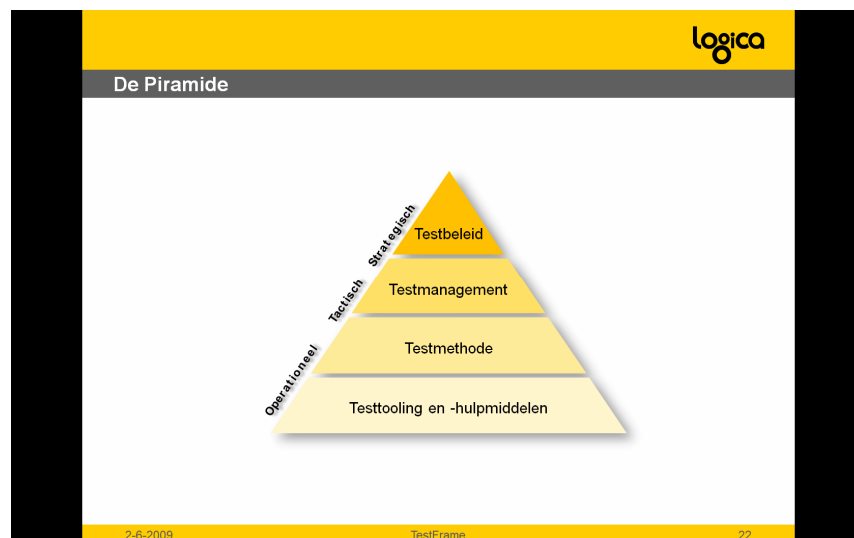
logica

Kortom

Met het nieuwe boek geven we TestFrame een nieuwe start. TestFrame blijft een moderne methode voor gestructureerd testen;

- Die naadloos aansluit bij RRBT (Risk & Requirement Based Testen)
- Waarmee de activiteiten binnen een testsoort of testtype worden gestructureerd
- Waarbij de testvoorbereiding en –uitvoering zijn gescheiden
- Waarbij de gevoeligheid voor onderhoud minimaal is en kennisborging maximaal is
- Die het mogelijk maakt om testgevallen te laten uitvoeren door medewerkers zonder testkennis
- Die het mogelijk maakt om testgevallen geautomatiseerd te laten uitvoeren
- Die voor alles de testmanager en stakeholders inzicht verschaft in de kwaliteit van het testobject en de voortgang van het testproces

2-6-2009 TestFrame 21



logica

Meer informatie

www.TestFrame.com
chris.schotanus@logica.com

2-6-2009 TestFrame 24