

TestGoal

TestGoal

Leerboek resultaatgericht software testen

Derk-Jan de Groot

Meer informatie over deze en andere uitgaven kunt u verkrijgen bij:

Sdu Klantenservice

Postbus 20014

2500 EA Den Haag

tel.: (070) 378 98 80

fax: (070) 378 97 83

© 2008 Sdu Uitgevers bv, Den Haag
Academic Service is een imprint van Sdu Uitgevers bv

1e druk, mei 2008

Zetwerk: Redactie bureau Ron Heijer, Markelo
Omslagontwerp: A-Graphic Design, Apeldoorn

ISBN: 978 90 395 2561 6
NUR: 123/980

Alle rechten voorbehouden. Alle auteursrechten en databankrechten ten aanzien van deze uitgave worden uitdrukkelijk voorbehouden. Deze rechten berusten bij Sdu Uitgevers bv.

Behoudens de in of krachtens de Auteurswet 1912 gestelde uitzonderingen, mag niets uit deze uitgave worden veelelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand of openbaar gemaakt in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voorzover het maken van reprografische veelelvoudingen uit deze uitgave is toegestaan op grond van artikel 16 h Auteurswet 1912, dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (Postbus 3051, 2130 KB Hoofddorp, www.reprorecht.nl). Voor het overnemen van gedeelte(n) uit deze uitgave in bloemlezingen, readers en andere compilatiewerken (artikel 16 Auteurswet 1912) dient men zich te wenden tot de Stichting PRO (Stichting Publicatie- en Reproductierechten Organisatie, Postbus 3060, 2130 KB Hoofddorp, www.cedar.nl/pro). Voor het overnemen van een gedeelte van deze uitgave ten behoeve van commerciële doeleinden dient men zich te wenden tot de uitgever.

Hoewel aan de totstandkoming van deze uitgave de uiterste zorg is besteed, kan voor de afwezigheid van eventuele (druk)fouten en onvolledigheden niet worden ingestaan en aanvaarden de auteur(s), redacteur(en) en uitgever deswege geen aansprakelijkheid voor de gevolgen van eventueel voorkomende fouten en onvolledigheden.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the publisher's prior consent.

While every effort has been made to ensure the reliability of the information presented in this publication, Sdu Uitgevers neither guarantees the accuracy of the data contained herein nor accepts responsibility for errors or omissions or their consequences.

Voorwoord

Voor je ligt het eerste *speciaal voor het hoger onderwijs* geschreven boek over software testen. Dit is dus niet het eerste boek over testen. Er zijn veel boeken beschikbaar die vertellen wat testen is, hoe dit zou moeten gebeuren en waarom dit belangrijk is. Waarom dan dit boek? Een voor de hand liggende vraag, die je als kritische student mag stellen. Er zijn een aantal punten te noemen waarop dit leerboek zich positief onderscheidt.

Het boek is primair gericht op de student. Het is geschreven in duidelijke taal. Het is erop gericht studenten concrete handvatten te geven om zelf te kunnen testen. Concrete voorbeelden en oefeningen helpen hen daarbij de testactiviteiten invulling te geven en de vraag te beantwoorden, *wat is testen?*

TestGoal omschrijft niet alleen de testactiviteiten. Er wordt uitgebreid stil gestaan bij de rol die testen speelt in de organisatie en binnen het software-ontwikkeltraject. Hierdoor krijgt de student inzicht in de toepassing van de geleerde technieken en tools en inzicht in de toegevoegde waarde die het testen heeft. Hij is in staat de vraag te beantwoorden, *waarom testen?*

Het is gebruikelijk in de literatuur te omschrijven hoe testen zou moeten gebeuren. Maar de werkelijkheid is weerbarstiger, en in veel organisaties wordt anders getest dan de methodieken voorschrijven. Natuurlijk is het belangrijk om als student de theorie te kennen en de technieken te kunnen toepassen. TestGoal gaat hier niet aan voorbij. TestGoal staat echter ook stil bij alternatieven. Het geeft antwoord op de vraag, *hoe kan het anders?*

Succes zit hem niet de gebruikte methodiek, maar in de wijze waarop de methodiek wordt toegepast. Dit leerboek richt zich daarom sterk op de persoon achter de tester. Wat onderscheidt de supertester van de middelmatige? TestGoal laat zien dat dit meer is dan een certificaat en beantwoordt de vraag, *hoe wordt ik een resultaatgedreven tester?*

Tot slot, testen wordt door veel studenten gezien als een onaantrekkelijk vak. In veel ogen is testen is een specialisatie die doorgroei beperkt. Als tester moet je vaak repeterend werk doen. Als tester kom je inhoudelijk minder uitdagingen tegen. Daarnaast is het maken van een product altijd beter dat het vinden van fouten er in. Dit boek laat zien dat niets minder waar is. Het laat zien *waarom testen leuk is.*

Derk-Jan de Grood
Leiden, april 2008

Inhoud

Voorwoord	
Introductie	
Introductie software testen	
1	Inleiding software testen
1.1	Geschiedenis van het testen
1.2	Overzicht beschikbare methodieken
2	Resultaatgedreven testen
2.1	Beoogd resultaat van dit hoofdstuk
2.2	Het belang van IT-systemen
2.3	Een uitspraak over kwaliteit
2.4	Perceptie van het testen
2.5	Een gezamenlijk doel
2.6	Aansluiten bij de business
2.7	Resultaatgedreven testen
2.8	Focus op resultaat
2.9	Zelftoets
2.10	Opdrachten
3	Testgoal en de tien testprincipes
3.1	Beoogd resultaat van dit hoofdstuk
3.2	Testprincipes
3.3	Focus op resultaat
3.4	Bouw aan vertrouwen
3.5	Neem verantwoordelijkheid
3.6	Beheers het testvak
3.7	Test gefaseerd
3.8	Faciliteer de gehele IT-lifecycle
3.9	Sla bruggen
3.10	Geef overzicht en inzicht
3.11	Zorg voor herbruikbaarheid
3.12	Bedenk: testen is leuk
3.13	Toepassen van de testprincipes
3.14	Zelftoets
3.15	Opdrachten
4	Testexpertise
4.1	Beoogd resultaat van dit hoofdstuk
4.2	De testfuncties
4.3	Carrièremogelijkheden
4.4	Zelftoets
4.5	Opdrachten
5	De aanpak
5.1	Beoogd resultaat van dit hoofdstuk
5.2	Context van het testtraject
5.3	Testsoorten en het V-model
5.4	Invulling geven aan het testtraject
5.5	Testen in een nieuwbouwproject
5.6	Testen in een beheeromgeving
5.7	Testen van conformiteit en interoperabiliteit
5.8	Testen van performance
5.9	Zelftoets
5.10	Opdrachten
6	Toepassen van het stappenplan
	Stap 1: Resultaat
7	Inventarisatie beoogd resultaat
7.1	Beoogd resultaat van dit hoofdstuk
7.2	Inleiding
7.3	Doel van de inventarisatie
7.4	Zelftoets
7.5	Opdrachten
	Stap 2: Aanpak
8	Testrisicoanalyse
8.1	Beoogd resultaat van dit hoofdstuk
8.2	Introductie
8.3	De 1-D testrisicoanalyse
8.4	De 2-D testrisicoanalyse

8.5	Zelftoets	12.8	Zelftoets
8.6	Opdrachten	12.9	Opdrachten
9	Testbegroting en -planning	13	Het fysieke testontwerp
9.1	Beoogd resultaat van dit hoofdstuk	13.1	Beoogd resultaat van dit hoofdstuk
9.2	Inleiding	13.2	Inleiding
9.3	Testplanning	13.3	Relatie met TRA en Logisch testontwerp
9.4	Zelftoets	13.4	Fysiek testgeval
9.5	Opdrachten	13.5	Testacties
10	Testplan	13.6	Zelftoets
10.1	Beoogd resultaat van dit hoofdstuk	13.5	Opdrachten
10.2	Inleiding	14	Testdata
10.3	Opdrachtformulering	14.1	Beoogd resultaat van dit hoofdstuk
10.4	Testbasis	14.2	Introductie
10.5	Teststrategie	14.3	Testdata-soorten
10.6	Testorganisatie	14.4	Testdata repository
10.7	Wijzigingen en afwijkingen	14.5	Productiedata versus testdata
10.8	Zelftoets	14.6	Het opnemen van data in het fysieke testontwerp
10.9	Opdrachten	14.7	Geautomatiseerde testuitvoering
Stap 3: Ontwerp		14.8	Testdata en exploratory testen
11	Intake testbasis	14.8	Back-up en restore
11.1	Beoogd resultaat van dit hoofdstuk	14.9	Zelftoets
11.2	Inleiding	16.10	Opdrachten
11.3	Invullen van de checklist Intake testbasis	Stap 4: Inrichting	
11.4	Ervaring verwerken in de checklist	15	Testomgeving
11.5	Review testbasis	15.1	Beoogd resultaat van dit hoofdstuk
11.6	Zelftoets	15.2	Inleiding
11.7	Opdrachten	15.3	Vaststellen Eisen testomgeving
12	Logisch testontwerp	15.4	Inrichten Realiseren testomgeving
12.1	Beoogd resultaat van dit hoofdstuk	15.5	Configureren en Intake systeem
12.2	Introductie	15.6	Zelftoets
12.3	Testontwerptechnieken	15.7	Opdrachten
12.4	Slim toepassen van testontwerptechnieken	16	Testautomatisering
12.5	Weinig ervaring met testontwerptechnieken?	16.1	Beoogd resultaat van dit hoofdstuk
12.6	Geen testontwerptechnieken	16.2	Inleiding
12.7	Toepassen testontwerptechnieken	16.3	Wat is Testautomatisering?

16.4	Dynamische testtools	20	Testrapportage
16.5	Statische testtools	20.1	Beoogd resultaat van dit hoofdstuk
16.6	Ondersteunende tools	20.2	Inleiding
16.7	Testautomatisering:Ja/Nee	20.3	Elementen in de testrapportage
16.8	Het ontwikkelen van testscripts	20.4	Het dashboard
16.9	Testautomatisering voor systemen met meer interfaces	20.5	Helderheid van de testrapportage
16.10	Zelftoets	20.3	Zelftoets
16.11	Opdrachten	20.4	Opdrachten
17	Intakesysteem		Stap 6: Borging
17.1	Beoogd resultaat van dit hoofdstuk	21	Borging
17.2	Inleiding	21.1	Beoogd resultaat van dit hoofdstuk
17.3	Invullen van de checklist	21.2	Evaluatie testtraject
17.4	Beheer van de checklist	21.3	Vaststellen Regressietestset
17.5	Zelftoets	21.4	Het archiveren en borgen van testware
17.6	Opdrachten	21.5	Overdracht
		21.6	Decharge van het testteam
		21.3	Zelftoets
		21.4	Opdrachten
			Dankwoord
			Over de auteur
			Bijlage A Functionele specificatie t.b.v. de CCS case
			Bijlage B Templates
			Bijlage C Woordenlijst
			Bijlage D Referenties
			Register
Stap 5: Uitvoering			
18	Testuitvoering		
18.1	Beoogd resultaat van dit hoofdstuk		
18.2	Testuitvoering		
18.3	De testuitvoering en zijn activiteiten		
18.4	Activiteiten tijdens de testuitvoering		
18.5	Testrun en regressietesten		
18.6	Buiten de geëffende paden treden		
18.7	Wanneer klaar met testen ?		
18.8	Zelftoets		
18.9	Opdrachten		
19	Bevindingenregistratie en -beheer		
21.1	Beoogd resultaat van dit hoofdstuk		
21.2	Inleiding		
21.3	Invullen van de bevindingenregistratie		
21.4	Attributen van een bevinding		
21.5	Bevindingenbeheer		
21.3	Zelftoets		
21.4	Opdrachten		

Introductie

Didactische verantwoording

Bij het schrijven van dit leerboek voor het hoger onderwijs is het uitgangspunt gehanteerd dat de student zich een goed beeld moet kunnen vormen van het vak software testen. Het boek bevat daarom nuttige, in het werkveld beproefde, theorie. Maar kennis alleen is niet voldoende. Bij het leren gaan lezen, denken en doen hand in hand. Elk hoofdstuk bevat naast de nodige theorie, een zelftoets die gebruikt kan worden om te toetsen of de student de theorie goed gelezen heeft. Opdrachten zorgen ervoor dat de student de theorie kan toepassen. Daarnaast wordt hij aan de hand van de opdrachten uitgedaagd om met zijn mede studenten de discussie aan te gaan en zodoende niet alleen kennis, maar ook begrip op te bouwen.

Dit leerboek is zodanig opgezet dat het op verschillende manieren gebruikt kan worden:

Als ondersteuning bij de colleges

Elke belangrijke activiteit in en product van het testproces is in een apart hoofdstuk behandeld. Dit maakt het voor de docent en student gemakkelijk om die passages te vinden die aansluiten bij de in het college behandelde stof. Elk hoofdstuk is voorzien van vragen en opgaven die beantwoord kunnen worden zonder de overige hoofdstukken te hoeven kennen. Binnen het hoofdstuk worden echter ook de relaties aangegeven met andere activiteiten in het testproces. Dit vergroot het inzicht en maakt verder studeren mogelijk, zowel in groepsverband als individueel.

Als integrale lesmethode voor een minor gewijd aan software testen.

Het boek doorloopt alle belangrijke activiteiten en hun bijbehorende producten in een logische volgorde. Hierdoor is het boek zeer geschikt als integrale methode voor een minor gewijd aan software testen. De student krijgt inzicht in de activiteiten die nodig zijn en hun onderlinge relatie, en krijgt een schat aan praktische tips en handvatten. De opdrachten stellen de student in staat ook daadwerkelijk te oefenen met de materie. Dit kan onder meer gedaan worden aan de hand van de integrale case die door het hele boek loopt. Inzichtvragen stimuleren discussie zorgen voor kennis *en* begrip.

Als ondersteuning bij projectonderwijs

Als studenten aan de hand van een project ervaring op doen met de testtheorie, wordt deze vaak vooraf gedoceerd en geoefend. De indeling van het boek maakt het voor de docent gemakkelijk de voorbereidende lesstof te vinden. De bijbehorende opgaven kunnen daarbij gebruikt worden voor het oefenen. Na bestude-

ring van de theorie en de daarin opgenomen voorbeelden, kan de student de verkregen kennis toepassen op de projectopdracht. Eventueel kan in het boek opgenomen integrale case gebruikt dienen projectcase op zich.

Als referentie bij stage, afstuderen of eerste baan

Door de vele praktische tips en de heldere structuur van het boek, leent het zich uitstekend als referentie op het moment dat studenten tijdens stage, afstudeeropdracht of bij hun eerste baan het testen van software in praktijk moeten brengen. De inhoud is getoetst aan de praktijk en de tester die TestGoal verantwoord toepast, voert een volwaardige testopdracht uit die niet onder hoeft te doen voor die van zijn collega's.

De bij dit boek behorende docentenhandleiding ondersteunt de docent in het toepassen van dit leerboek bij zijn colleges. Het bevat niet alleen de uitwerkingen van alle vragen en opdrachten, maar geeft ook achtergrondinformatie die de colleges extra diepgang kunnen geven.

Voor wie is dit boek bedoeld?

Dit boek richt zich op de studenten die een hbo-opleiding of academische opleiding volgen in de IT-richting. Velen van hen zijn weliswaar geïnteresseerd in een carrière als testexpert, maar vinden het moeilijk om in te schatten wat het vak precies inhoudt. Mede debet hieraan is de beperkte aandacht die het testvak krijgt tijdens de opleiding. Dit boek geeft aan hoe testen kan bijdragen aan de ontwikkeling van de organisatie en door zijn praktische invulling geeft het ook een helder beeld van wat testen nu eigenlijk inhoudt. Daarnaast hoop ik dat het plezier dat ik aan het vak beleef – en dat naar ik hoop tussen de regels door te lezen is – hen ook enthousiast kan maken voor het mooie vak van tester.

Meer weten?

TestGoal is een filosofie met een praktijkgerichte aanpak die wordt ondersteund door een aantal praktische templates. Deze templates zijn samen met de aanpak ontwikkeld en maken het inzetten van TestGoal eenvoudiger en efficiënter. Deze templates zijn niet in dit boek opgenomen. U kunt deze vinden op de speciale website: www.testgoal-educatief.nl.

Hoofdstuk 8

Testrisicoanalyse

Na het bestuderen van dit hoofdstuk is de student in staat

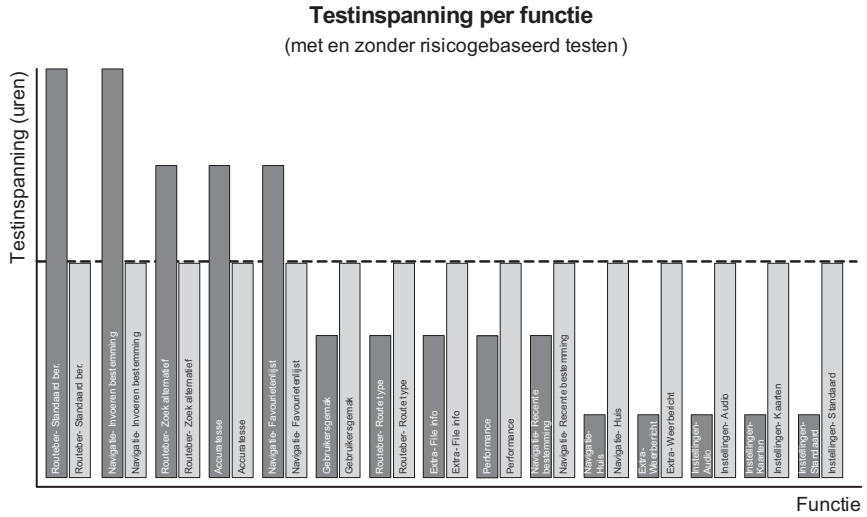
- het basisprincipe van risicogebaseerd testen uit te leggen;
- aan te geven wanneer de testrisicoanalyse (TRA) wordt gebruikt in het testtraject en met welk doel;
- het verschil uit te leggen tussen 1-D en 2-D TRA;
- een testrisicoanalyse uit te voeren.

8.1 Introductie

Er zijn vele bekende voorbeelden die voorrekenen dat het onmogelijk is om alles te testen. Zelfs bij kleine functies is het aantal mogelijkheden al zo groot, dat zelfs als we alle mogelijke testen heel snel (geautomatiseerd) zouden uitvoeren, de testuitvoering toch nog ondoenlijk lang zou duren. Met de toenemende complexiteit en het steeds grotere aandeel dat software heeft in de dagelijkse producten, wordt het kwaliteitsprobleem steeds groter. De hoeveelheid software in consumentenelektronica verdubbelt elk jaar, terwijl het de foutintensiviteit, het aantal fouten per 1000 regels code (KLOC), de afgelopen 10 jaar niet echt verminderd is [Veenendaal, 2007]. Tel daar bij op, dat de time-to-market van softwareproducten steeds korter wordt en de uitdaging is compleet. Hoe gaan we hier als testers mee om? Het antwoord is simpel, we testen slechts delen van de software goed, maar denken wel heel goed na over wat we minder goed of helemaal niet testen. Deze gedachte vormt de basis voor 'risicogebaseerd testen'.

Figuur 8.1 geeft het principe van risicogebaseerd testen weer aan de hand van functies van een simpel routenavigatiesysteem. Wordt er niet risicogebaseerd getest, dan zullen alle functies evenveel aandacht krijgen. Laten we aannemen dat er per functie vier uur wordt getest, dan zal de totale inspanning $15 * 4$ uur = 60 uur zijn. Wordt risicogebaseerd testen geïntroduceerd, dan wordt de testinspanning gevarieerd per functie. De onbelangrijkere functies worden slechts een beetje getest. De tijd die hiermee uitgespaard wordt, kan worden gebruikt om extra aandacht te geven aan de belangrijkste functies. Voordeel is dat de tijd en aandacht gaan zitten in die zaken die toegevoegde waarde hebben voor het beoogde resultaat [Pinkster et al, 2004]. Idealiter worden hierdoor de belangrijke functies aanzienlijk beter getest dan voorheen het geval was, en is de totale testinspanning kleiner dan de initiële 60 uur. De testrisicoanalyse (TRA) is het instrument om vast te stellen welke functies belangrijk zijn, en welke functies minder goed getest hoeven te worden.

testrisicoanalyse
(TRA)



Figuur 8.1 Bij risicogebaseerd testen wordt de testinspanning gevarieerd. Voor het testen van de belangrijke functies wordt meer tijd gereserveerd dan in de situatie waarbij niet risicogebaseerd getest wordt. In deze situatie is de testinspanning voor elk van de functies gelijk.

De TRA wordt daarnaast gebruikt voor de volgende zaken:



- De TRA ondersteunt de keuze voor de toe te passen testontwerptechnieken.
- Bij reviews en intakes kan op basis van de TRA worden besloten belangrijke onderdelen beter te reviewen dan de minder belangrijke.



- Tijdens de testuitvoering wordt de TRA gebruikt om de volgorde vast te stellen waarin de testen uitgevoerd worden. Het is gebruikelijk om te beginnen met de belangrijkste testen. Dit vergroot de kans dat de belangrijkste bevindingen snel worden gevonden. Daarnaast heeft dit het voordeel dat als de testuitvoering vroegtijdig wordt afgebroken, de belangrijke testen zijn uitgevoerd.



- In de testrapportage wordt verwezen naar de risico's. Dit heeft als voordeel dat de testrapportage verhaalt over zaken die de belanghebbenden aanspreken, namelijk de risico's voor het beoogde resultaat.

We onderscheiden twee vormen van TRA die naast elkaar gebruikt kunnen worden. Hoe deze opgesteld kunnen worden, wordt behandeld in de volgende paragrafen.

8.2 De 1-D testrisicoanalyse

Bij de eendimensionale TRA wordt het relatieve belang bepaald van de verschillende onderdelen van het testobject. Aan de hand van een functionele decompositie wordt een overzicht gemaakt van de functies die het systeem moet kunnen ondersteunen. Input voor de 1-D TRA is de systeem- of requirementsspecificatie. Hierin zijn de functies benoemd die het systeem moet ondersteunen. Het

1-D TRA

product van de 1-D TRA is een eendimensionale risicomatrix. Dit is een lijstje met de functies, gerangschikt naar hun relatieve belang. De 1-D TRA is goed bruikbaar om per functie de testdiepte te bepalen en om de volgorde van de testuitvoering te bepalen.

De Testrisicoanalyse wordt uitgevoerd tijdens een workshop waarbij verschillende belanghebbenden aanwezig zijn. De belanghebbenden kunnen vanuit hun expertise en werkveld de functies en aandachtsgebieden identificeren en van een prioriteit voorzien. De TRA wordt georganiseerd door de moderator die hierbij als procesbegeleider fungeert. Hij is degene die de TRA-sessie organiseert en na afloop de gegevens verwerkt. Vaak fungeert de testcoördinator als moderator.

Bij het tot stand komen van de eendimensionale TRA doorlopen we de volgende stappen.

- 1 Identificeren belanghebbenden en kick-off.
- 2 Vaststellen van de functies en aandachtsgebieden.
- 3 Bepalen van het relatief belang.
- 4 Verwerken van de gegevens.
- 5 Accorderen van de TRA.

In de volgende paragrafen worden de stappen nader toegelicht.

8.2.1 Identificeren belanghebbenden en kick-off

De in de ‘Inventarisatie beoogd resultaat’ geïdentificeerde belanghebbenden worden door de moderator uitgenodigd voor de TRA-sessie. De verschillende belanghebbenden hebben verschillende visies op het beoogde resultaat en de risico's [Thompson, 2004]. Het betrekken van verschillende disciplines helpt om een evenwichtige en doordachte risicoinschatting te krijgen. Denk aan:

- De afnemers (gebruikers, businessmanager of beheerders).
- De bouwers (analisten, systeemontwerpers of programmeurs)
- Projectverantwoordelijken (projectmanagers of opdrachtgever).



kick-off

Tijdens een kick-off legt uit de moderator uit waarom de TRA gehouden wordt, hoe dit in zijn werk gaat en wat er van de belanghebbenden wordt verwacht. Het is belangrijk dat elke deelnemer aan de TRA deskundig genoeg wordt bevonden door de groep die hij vertegenwoordigt. Dit zorgt er voor dat de deelnemer onderbouwde uitspraken kan doen over de prioriteit. Het zorgt er ook voor dat zijn inschatting door zijn achterban wordt geaccepteerd. Dit lijkt misschien een klein issue, maar dit is toch belangrijk, bijvoorbeeld in het volgende geval.

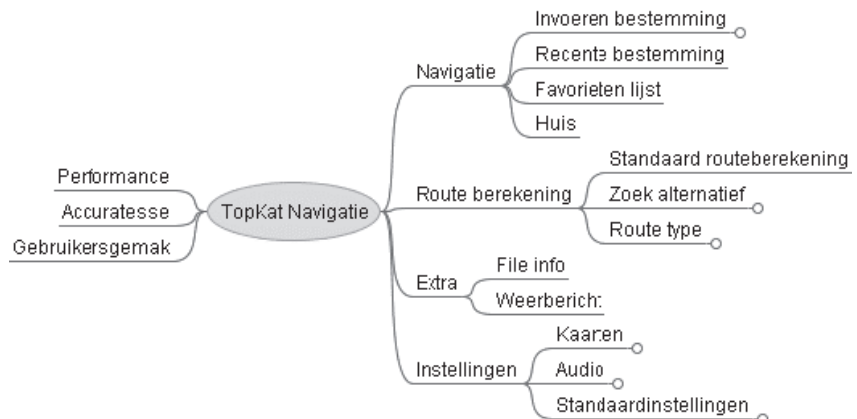
Voorbeeld 8.1: Een webapplicatie

Binnen de organisatie wordt een webapplicatie gebouwd waarmee klanten bestellingen kunnen plaatsen. Tijdens de TRA geeft de projectleider aan dat er weinig tijd beschikbaar is. Hierdoor zullen er concessies moeten worden gedaan aan de opgeleverde functionaliteit. De deelnemers besluiten samen dat het deel dat de klanten te zien krijgen niet gecompromitteerd mag worden. De webinterface dient met zorg ontwikkeld en getest te worden. De applicatie heeft ook nog een andere interface. Dit zijn de schermen waarmee de medewerkers van de orderafhandeling moeten werken. Tijdens de TRA wordt besloten dat deze schermen functioneel moeten werken, maar dat de gebruikersvriendelijkheid niet getest hoeft te worden. De gebruikersvertegenwoordiger weet dat zijn achterban niet blij is met deze beslissing. Doordat deze interface minder goed getest wordt, is de kans groot dat hun werk wordt gehinderd. Toch stemt hij in met de beslissing. Het lijkt de meest verstandige beslissing, en hij weet dat de gebruikers zijn beslissing zullen respecteren.

8.2.2 Vaststellen van de functies en aandachtsgebieden

Testboom

Voordat de moderator de risicogebieden kan prioriteren, dient hij deze in kaart te brengen en stelt hij hiertoe een Testboom op. Een testboom is een mind-map die aangeeft welke functies en aandachtspunten een systeem heeft. Elke functie of aandachtsgebied wordt als een tak in de boom getekend. Eventueel kunnen functies en aandachtsgebieden nog uiteengehaald worden. De tak splitst zich in dat geval op in een of meer takjes.



Figuur 8.2 Testboom opgesteld voor een eenvoudig navigatiesysteem

In figuur 8.2 is een Testboom uitgewerkt voor een eenvoudig navigatiesysteem. In de Testboom is een aantal hoofdfuncties te herkennen: Navigatie, Route berekenen, Extra en Instellingen. Elk van de hoofdfuncties, ook wel functie-groepen genoemd, is een verzameling van functies. De hoofdfunctie Navigatie

verzamelt alle manieren waarop de eindbestemming van de reis ingevoerd kan worden. Zoals is te zien in de Testboom kan dit op een aantal manieren: door een nieuwe bestemming in te voeren, of door een bestemming te selecteren uit de lijst met recente of favoriete bestemmingen. Ook is het mogelijk om naar Huis te worden geleid. Aan de linkerkant in de Testboom zijn enkele niet-functionele aandachtsgebieden weergegeven, die ook uiteengerafeld kunnen worden. Accuratesse bijvoorbeeld kan gesplitst worden in het berekenen van de reistijd, het op het juiste moment aangeven van een afslag of, niet onbelangrijk, het correct aangeven van de werkelijk snelste route.

prioritering

Het gebruik van de Testboom als basis voor de Testrisicoanalyse heeft een aantal voordelen. De Testboom vormt tijdens de TRA een checklist die ervoor zorgt dat er geen functies worden vergeten en is daarna te gebruiken als structuur voor het fysieke testontwerp. Tijdens de testuitvoering kunnen hierdoor de testresultaten gemakkelijk teruggeleid worden naar de Testrisicoanalyse. Dit maakt het mogelijk om over risico's te rapporteren naar de belanghebbenden.

De functies in de Testboom worden afgeleid uit de testbasis (bijvoorbeeld omschreven functies uit de requirementsspecificatie, functioneel ontwerp en use cases). De aandachtsgebieden worden in kaart gebracht tijdens interviews met de belanghebbenden. Dit kan tijdens de TRA-sessie, maar kan ook op voorhand. Een goede voorbereiding zorgt ervoor dat de eigenlijke TRA-sessie niet te lang gaat duren. Tijdens de TRA-sessie controleren de deelnemers of de Testboom compleet is en kunnen zij eventuele nieuwe inzichten aan de Testboom toevoegen. Aanvullend zorgt het langslopen van de kwaliteitsattributen uit het Quint-model (het Extended ISO 9126-model) ervoor dat ook de non-functionele aspecten niet vergeten worden.

Als de functies en aandachtsgebieden in kaart zijn gebracht, dan wordt het tijd om het relatieve belang van de risicogebieden te bepalen.

8.2.3 Bepalen van het relatief belang

Een efficiënte benadering is om alle belanghebbenden te vragen de takken van de Testboom te voorzien van een prioritering. Elk van de belanghebbenden verdeelt zijn punten over de takken van de Testboom. Dit kan door middel van individuele interviews of, als de groep niet te groot is, tijdens een plenaire sessie. Nadat iedereen zijn input heeft gegeven, worden deze ratings samengevoegd. De volgende kwalificaties worden gebruikt.

Testboom

9 punten

De functie of het aandachtspunt is van cruciaal belang voor de werking van het systeem en het behalen van het beoogde resultaat. Fouten in deze functie, of een ontoereikende implementatie, hebben een directe invloed op de bruikbaarheid van het systeem. Het systeem kan pas worden vrijgegeven als deze functie of dit aandachtsgebied grondig getest is.

5 punten

Het betreft een belangrijke functie. Fouten in deze functies, of een ontoereikende implementatie, zijn toegestaan als er een workaround beschikbaar is. Functies en aandachtsgebieden dienen goed getest te worden alvorens ze vrijgegeven kunnen worden.

3 punten

Niet-cruciale functie of aandachtsgebied. Fouten in de functie kunnen hinderlijk zijn, maar brengen naar verwachting het beoogde resultaat niet in gevaar. Functies en aandachtsgebieden dienen getest te worden alvorens ze vrijgegeven kunnen worden.

1 punt

Functie of aandachtsgebied dat niet noodzakelijk is voor de werking van het systeem. Deze worden bij voorkeur getest, maar slechts met een beperkte diepgang.

Bij het toekennen van punten is het belangrijk om een goede verdeling na te streven over de risicogebieden. Deelnemers die nog niet zo ervaren zijn met risicoinschattingen hebben de neiging om alles belangrijk te vinden. Benadruk dan dat het belangrijk is om onderscheid te maken. Leg de toepassing van de Testrisicoanalyse nogmaals uit en benadruk dat het om een relatief belang gaat. Misschien is inderdaad alles belangrijk, maar zelfs dan zijn sommige zaken belangrijker dan andere.

In tabel 8.1 zijn de gegevens van een TRA-sessie met vier deelnemers weergegeven.

Tabel 8.1 Gegevens van een TRA-sessie

Risicogebied	Verdeling	Verdeling	Verdeling	Verdeling	Totaal
	1	2	3	4	
Navigatie-Invoeren bestemming	9	9	3	9	30
Navigatie-Recente bestemming			3	1	4
Navigatie-Favorietenlijst	9	1	3	3	16
Navigatie-Huis			3		3
Routeber.-Standaard ber.	9	9	3	9	30
Routeber.-Zoek alternatief		5	3	5	13
Roeteber.-Route type		1	3	3	7
Extra-File info		3	3	9	15
Extra-Weerbericht			3		3
Instellingen-Kaarten			3	3	6
Instellingen-Audio		5	3		8
Instellingen-Standaard		3	3		6
Performance	9	3	3		15
Accuratesse		5	3	3	11
Gebruikersgemak	9	1	3		13
Totaal	45	45	45	45	

Mocht het niet lukken onderscheid te maken tussen de functies en aandachtspunten, geef dan de belanghebbenden een gelimiteerd aantal punten. In het bovenstaande voorbeeld is dit toegepast. De deelnemers aan de sessie hebben maximaal 45 punten te verdelen over 12 risicogebieden. Er kunnen maar 5 x 9 punten worden uitgedeeld (verdeling 1). Wil de deelnemer meer dan vijf risicogebieden aanwijzen, dan wordt hij gedwongen ook minder hoge punten toe te kennen (verdeling 2). Mocht een deelnemer toch al zijn punten gelijk willen verdelen, dan kan dat (zie verdeling 3). Echter doordat er een maximum aantal punten toe te kennen is, minimaliseert hij zijn invloed op het totaal. Door de deelnemer hierop te wijzen, zal hij waarschijnlijk toch besluiten een andere verdeling toe te kennen.

Het kan heel goed zijn, dat een functie die op zichzelf niet zo belangrijk is, wel een aantal paden bevat die belangrijk zijn. Dit blijkt duidelijk als niet alleen het laagste niveau van de Testboom een relatief belang krijgt, maar ook het niveau daarboven. In het onderstaande voorbeeld is een prioritering aangebracht per functie en per subfunctie. De functies en subfuncties hebben onafhankelijk van elkaar een prioriteit gekregen. Bij onderlinge vermenigvuldiging hiervan valt op dat de relatieve belangen van de subfuncties door elkaar gaan lopen. In tabel 8.2 is het relatieve belang van beide functies meegenomen in de totaalberekening.

Wat opvalt, is dat functie Extra veel minder belangrijk wordt gevonden dan de functie Navigatie, respectievelijk 3 en 9 punten. Toch is het relatieve belang van Extra-File-info groter dan van Navigatie-Recente bestemming. Houd er bij het opstellen van de TRA dus rekening mee dat ook een minder belangrijke functie toch belangrijke aspecten kan bevatten.

Tabel 8.2 *Het relatieve belang van functies*

Risicogebied	Relatief belang Functie	Relatief belang Sub-functie	Totaal
Routeber.-Standaard ber.	9	30	270
Navigatie-Invoeren bestemming	5	30	150
Routeber.-Zoek alternatief	9	13	117
Accuratesse	9	11	99
Navigatie-Favorietenlijst	5	16	80
Gebruikersgemak	5	13	65
Roeteber.-Route type	9	7	63
Extra-File info	3	15	45
Performance	3	15	45
Navigatie-Recente bestemming	5	4	20
Navigatie-Huis	5	3	15
Extra-Weerbericht	3	3	9
Instellingen-Audio	1	8	8
Instellingen-Kaarten	1	6	6
Instellingen-Standaard	1	6	6

8.2.4 Verwerken van de gegevens

De gegevens van elk van de deelnemers worden verzameld en gesorteerd naar relatief belang. Dit kan in heel goed in Excel worden gedaan. Vervolgens worden de risicogebieden samengevoegd in risicocategorieën. Deze risicocategorieën vormen het eindproduct van de Testrisicoanalyse. Tabel 8.3 geeft aan hoe de risicogebieden verdeeld kunnen worden over de risicocategorieën.

Tabel 8.3 *Risicocategorieën*

Risicocategorie	Inhoud
Kritisch	Belangrijkste 10 % van de risicogebieden
Hoog	Daaropvolgende 20% van de risicogebieden
Midden	Daaropvolgende 30% van de risicogebieden
Laag	Minst belangrijke 40% van de risicogebieden

Voor het voorbeeld uit de vorige paragraaf levert dit de volgende TRA op:

Op <datum> is een risicoanalyse uitgevoerd door:
 Naam, functie
 Naam, functie

De volgende risico's met hun relatief belang zijn erkend:

Risicocategorie	Risicogebied	Relatief belang
Kritisch	Routeber.-Standaard ber.	270
	Navigatie-Invoeren bestemming	150
Hoog	Routeber.-Zoek alternatief	117
	Accuratesse	99
	Navigatie-Favorietenlijst	80
Midden	Gebbruikersgemak	65
	Roeteber.-Route type	63
	Extra-File info	45
	Performance	45
	Navigatie-Recente bestemming	20
Laag	Navigatie-Huis	15
	Extra-Weerberich	9 t
	Instellingen-Audio	8
	Instellingen-Kaarten	6
	Instellingen-Standaard	6

Figuur 8.3 Een voorbeeld-TRA

8.2.5 Accorderen van de TRA

Als laatste stap moet overeenstemming worden bereikt over de TRA. Na het verwerken van de inschattingen en het toekennen van de risicocategorieën, is het goed de deelnemers en belanghebbenden om een akkoord te vragen. Vraag hen om na te gaan of er geen aandachtsgebieden onlogisch hoog of laag in de TRA zijn terechtgekomen. Alle op- en aanmerkingen kunnen in dit stadium nog zonder veel problemen worden verwerkt. Het is beter nieuwe inzichten nu te verwerken, dan later in het project. Wanneer de TRA is goedgekeurd, kan worden overgegaan tot het kiezen van de Teststrategie. Dit is omschreven in de volgende hoofdstukken.

8.3 De 2-D testrisicoanalyse

2-D TRA Het identificeren van de risico's in de 2-D TRA gebeurt op een andere manier dan bij de 1-D TRA. Het grote verschil tussen beide vormen in risicoanalyse is dat de 1-D TRA gebaseerd is op de systeem- of requirementsspecificatie. De 2-D TRA staat hier los van en gaat uit van risico's die het beoogde resultaat bedreigen. Met de 2-D TRA zullen daarom risico's gevonden worden die niet in de testbasis zijn geadresseerd. De 2-D TRA is goed bruikbaar om te bepalen welke testen er uitgevoerd moeten worden. Dit zal ook vaak leiden tot het testen van de non-functionele kwaliteitsattributen.



Nadat het beoogde resultaat is vastgesteld in de Resultaatsomschrijving, wordt onderzocht welke bedreigingen er zijn voor het beoogde resultaat. Bedreigingen zijn alle gebeurtenissen die ertoe kunnen leiden dat het resultaat niet wordt gehaald. Omdat niet alle bedreigingen even serieus genomen hoeven te worden, wordt het risico van elke gebeurtenis ingeschat. De volgende definitie van risico wordt gebruikt:

risico *Het risico van een gebeurtenis is het product van de kans dat de gebeurtenis optreedt met de impact die dit heeft op het resultaat. Of: Risico = Kans \times Impact.*

Binnen deze definitie is het resultaat opgenomen. Binnen het resultaatgedreven testen leiden we de geïdentificeerde risico's terug tot hun effect op het beoogde resultaat. Dit lichten we toe aan de hand van het Connecta-project (zie voorbeeld 1.3)

Voorbeeld 8.2: Risico en resultaat bij Connecta

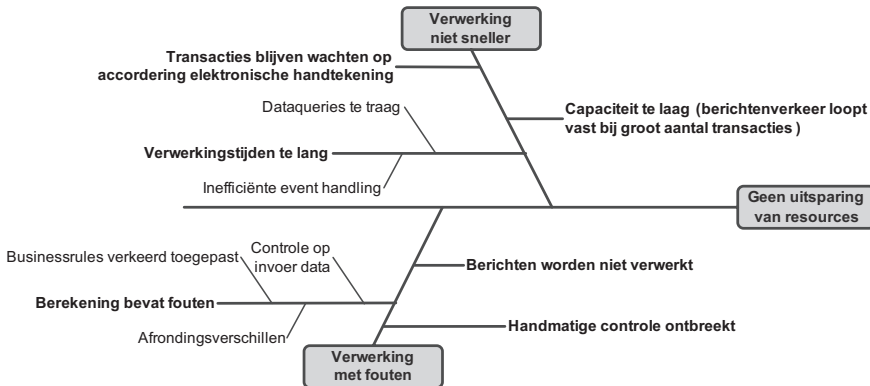
In de Resultaatsomschrijving voor de systeemtest van Connecta was het beoogde resultaat duidelijk geformuleerd. De business rekt erop dat het nieuwe systeem resources uitspaart. Door een aantal van de bestaande systemen te integreren gaat de verwerking niet alleen sneller, er worden ook minder fouten gemaakt. Nemen we dit beoogde resultaat als uitgangspunt, dan identificeren we meteen een risico. Immers, wat als blijkt dat er met het nieuwe systeem helemaal geen resources uitgespaard worden? Dit kan bijvoorbeeld het geval zijn als met het nieuwe geïntegreerde systeem:

- de verwerking niet sneller verloopt, maar even snel of langzamer;
- de verwerking gepaard gaat met evenveel of meer fouten.

De testen zullen een uitspraak moeten doen over de mate waarin de doelstellingen van het Connecta-project kunnen worden gehaald. Daarom zal de testcoördinator er minimaal voor moeten zorgen dat de testen een uitspraak doen over de snelheid waarmee het systeem zijn werk doet en de fouten die hierbij optreden.

Elk van deze risico's kan nader geanalyseerd worden. Het doel is te achterhalen welke gebeurtenissen het benoemde risico laten optreden. Vaak wordt hiervoor een Ishikawa-diagram of fishbone-diagram gebruikt (dit diagram wordt zo genoemd omdat de structuur lijkt op een visgraat) [Bilt]. In figuur 8.4 is zo'n diagram weergegeven.

fishbone-
diagram



Figuur 8.4 Een fishbone-diagram

In een fishbone-diagram worden oorzaken en gevolgen met elkaar in verband gebracht. In dit geval geeft het diagram gebeurtenissen weer die een risico vormen. Als bijvoorbeeld niet alle berichten worden verwerkt, bevat de output van de verwerking waarschijnlijk fouten. Als de verwerking fouten bevat, zal er binnen de organisatie handmatige correctie nodig zijn. Als risico was daarom ingeschat dat er in dat geval niet op de resources bespaard kan worden.

Nadat de risico's in kaart zijn gebracht, kunnen maatregelen worden genomen om de risico's aan te pakken. In de praktijk houdt dit in dat er een keuze gemaakt wordt uit de volgende mogelijkheden.

- Niet aanpakken van het risico.
- Onderzoeken in welke mate het risico reëel is.
- Wegnemen van de oorzaken die tot dit risico leiden.
- Reduceren van de gevolgen van risico.

Testen richt zich in de regel op de middelste twee punten. Door te testen wordt ervaring opgedaan met de werking van het systeem en ontstaat inzicht in de werkelijke kans en de impact van het risico. Door fouten te vinden en op te lossen, wordt de oorzaak van het risico weggenomen.

Voorbeeld 8.3: Afdekken van risico's in het Connecta-project

In het Connecta-project zijn onder meer de volgende twee risico's geïdentificeerd:

Risico 1: Een fout in de berekening zorgt voor veel handmatige follow-up.

Risico 2: De capaciteit van het systeem is onvoldoende en kan niet alle aanvragen tijdig verwerken.

Het testtraject richt zich op het afdekken van deze risico's op de volgende wijze:

Risico 1:

De berekening wordt getest aan de hand van realistische scenario's. De uitkomsten van elke stap worden vergeleken met de verwachte uitkomst. Aanvullend worden er ook onlogische combinaties van data ingevoerd om te controleren of de foutafhandeling adequaat werkt. Tijdens de testuitvoering is een aantal fouten gevonden dat ertoe leidde dat de berekening een verkeerde uitkomst gaf. Door deze fouten op te lossen, is de kans dat er in productie verkeerde berekeningen worden uitgevoerd aanzienlijk verkleind. Tijdens de testrapportage wordt dit testresultaat besproken. De belanghebbenden vinden dat risico 1 gereduceerd is tot een acceptabel niveau.

Risico 2:

Binnen de organisatie is bekend hoeveel aanvragen er per dag binnenkomen. Op basis van deze belasting wordt een performancetest uitgevoerd waarmee onderzocht wordt welke piekbelasting het systeem aankan. Bij hoge piekbelasting blijken berichten soms niet te worden verwerkt. Onderzoek leert echter dat men niet verwacht dat deze piek de komende jaren zal voorkomen. Vervolgens wordt er een duurttest uitgevoerd. De duurttest toont aan dat bij een belasting die 150% van het huidige aantal aanvragen telt, het systeem een week zonder problemen kan draaien. De test toont aan dat risico 2 wel aanwezig is, maar geen verdere aandacht behoeft.

Het uitvoeren van een 2-D TRA doorloopt de globaal dezelfde stappen als de 1-D TRA. De wijze waarop de deelnemers van de TRA worden geselecteerd, uitgenodigd en geïnformeerd is hetzelfde als bij de 1-D TRA (zie sectie 7.2.2). Het is raadzaam om na afloop ook de resultaten van de 2D-TRA te laten accorderen. Het vaststellen van de risico's gaat echter zoals beschreven in de volgende paragraaf.

8.3.1 Vaststellen van risico's

De deelnemers aan de TRA wordt gevraagd

- risico's te benoemen;
- te onderzoeken welke oorzaken een risico kan hebben;
- factoren te benoemen die van invloed zijn op de faalkans.

Bij de 2-D TRA worden voor alle vastgestelde bedreigingen de kans en de impact onafhankelijk van elkaar ingeschat. Dit gebeurt door elke factor die van invloed is op de faalkans en de impact punten te geven. Bij het inschatten van de impact wordt vastgesteld wat de mogelijke schade is als het risico optreedt. De mogelijke schade heeft betrekking op het beoogde resultaat en kan worden onderverdeeld naar bijvoorbeeld:

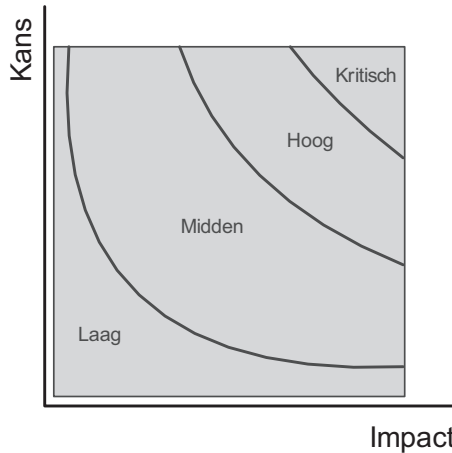
- financiën;
- imago;
- bedrijfsprocessen.

Bij het inschatten van de faalkans wordt ingeschat hoe groot de kans is dat het risico daadwerkelijk optreedt. Deze kans hangt af van de volgende factoren:

- Complexiteit van het onderdeel van het testobject.
- Omvang van het onderdeel van het testobject.
- Mate waarin er nieuwe onbeproefde technologieën worden gebruikt.
- Relaties met andere onderdelen.
- Kwaliteit van bouwteam.
- Frequentie waarmee het onderdeel wordt gebruikt.

8.3.2 Verwerken van de gegevens

Nadat in een risicoworkshop alle deelnemers hun input hebben geleverd, worden de impact en faalkans per risico uitgerekend. Het is gebruikelijk om de resultaten in een Kans-Impact grafiek uit te zetten. Net zoals bij de 1-D TRA worden de risico's toegekend aan risicocategorieën. Kritisch zijn die risico's waarvan impact en kans hoog zijn. De risico's met een kleine impact en een kleine kans behoren tot de risicocategorie Laag. Onderstaande grafiek geeft de risicocategorieën aan. [Baars et al, 2006], [Gardiner,2006], [Pinkster et al, 2004].



Figuur 8.5 De TRA-matrix. Het risico van een aandachtsgebied, uitgedrukt in geschatte faalkans en impact, kan worden weergegeven door een punt in de Kans-Impact Grafiek.

8.4 Zelftoets

1. Wat is risicogebaseerd testen?
2. Welke ontwikkelingen maken dat risicogebaseerd testen aan populariteit wint?
3. Noem twee voordelen van risicogebaseerd testen.
4. Wat zou het inhouden als er niet risicogebaseerd getest wordt?
5. Welke belanghebbenden worden er in de regel bij een TRA betrokken.
6. Wanneer zou je 1-D TRA toepassen?
7. Wat is een testboom? Hoe wordt deze gebruikt binnen de TRA?
8. Waarvoor wordt de testboom nog meer gebruikt?
9. Wanneer zou je 2-D TRA toepassen?
10. Wat is een fishbone-diagram. Waarvoor wordt dit gebruikt?
11. Bij welke activiteiten in het testtraject worden de resultaten van de TRA hergebruikt?

8.5 Opdrachten

In bijlage A is de functionele specificatie opgenomen van het Centrale Sanctie Systeem (CSS). De volgende opdrachten worden uitgevoerd aan de hand van deze specificatie. Indien mogelijk kan er worden voortgebouwd op de resultaten van opdrachten uit voorgaande hoofdstukken.

1. Maak een functionele decompositie voor het CSS uit de bijlage. Werk deze uit in een testboom.

2. Voer een 1D-risicoanalyse uit door aan elk van de takken van de in opdracht 1 opgestelde testboom een 'belang'toe te kennen. Noteer ook de afwegingen en aannames die je hebt gemaakt zodat je keuzes te onderbouwen zijn. Als je deze opdracht in een groep doet, verzamel dan alle individuele analyses en verwerk deze tot één gezamenlijke TRA. Bespreek onderling de verschillen tussen de individuele inschattingen. Worden deze verschillen veroorzaakt door verschillende aannames van het systeem of door verschillende achtergronden van de teamleden?
3. Bedenk drie situaties die ervoor zouden kunnen zorgen dat het CSS niet het resultaat oplevert dat de opdrachtgever nastreeft. Kies een situatie en onderzoek mogelijke gebeurtenissen die tot deze situatie leiden. Doe dit aan de hand van een fishbone-diagram.
4. Voor het fishbone-diagram opgesteld in opgave 3. Ken de kans en impact toe aan elk van de gebeurtenissen. Bereken voor elk van de gebeurtenissen het risico. Als je deze opdracht in een groep doet, verzamel dan alle individuele analyses en verwerk deze tot één gezamenlijke TRA. Bespreek onderling de verschillen tussen de individuele inschattingen. Worden deze verschillen veroorzaakt door verschillende aannames van het systeem of door verschillende achtergronden van de teamleden?
5. Zet voor/nadelen van 1D en 2D TRA tegen elkaar uit

	Voordeel	Nadeel
1D-TRA		
2D-TRA		

6. Verschillende belanghebbenden hebben waarschijnlijk verschillende visies op het beoogde resultaat en de risico's [Thompson, 2004]. Bedenk typische risico's die een businessmanager, een beheerder, een programmeur of een projectmanager zouden kunnen aandragen.
7. Bij het schatten van de faalkans wordt geschat hoe groot de kans is dat het risico daadwerkelijk optreedt. Deze kans hangt af van de volgende factoren:
 - Complexiteit van het onderdeel van het testobject.
 - Omvang van het onderdeel van het testobject.
 - Mate waarin er nieuwe onbeproeft technologieën worden gebruikt.
 - Relaties met andere onderdelen.
 - Kwaliteit van bouwteam.
 - Frequentie waarmee het onderdeel wordt gebruikt.

Geef voor elke van de factoren een voorbeeld of toelichting waarom dit de software kwaliteit beïnvloedt.

8. Als testcoördinator op een kortgeleden gestart testtraject, heb je jouw opdrachtgever verteld dat je een TRA wil gaan organiseren. Je opdrachtgever geeft aan dit niet echt te zien zitten. Het houden van zo'n TRA belast de rest van de organisatie en kost tijd. Tijd die beter besteed kan worden aan het opstellen van een grondig testdraaiboek. Welke argumenten geef jij hem, om hem te overtuigen van het nut van risicogebaseerd testen?
9. In paragraaf 5.7 worden conformiteit- en interoperabiliteitstesten behandeld. Welke rol speelt de TRA bij dit soort testen?

Over de auteur

Derk-Jan de Groot werkt sinds 1997 in het softwaretestveld. Als testmanager voor Collis geeft hij bedrijven advies bij het opzetten van testen, testprojecten en testcentra. Hij is gecertificeerd ISEB practitioner en is de afgelopen jaren betrokken geweest bij meerdere projecten in een groot aantal branches, waarbij hij regelmatig contact heeft gehad met gebruikers, beheerders, automatiseringsafdelingen en externe leveranciers.

Derk-Jan heeft zijn ervaring onder meer opgedaan als uitvoerend tester, waarbij inhoud en techniek centraal staan. Daarnaast heeft hij adviesprojecten verzorgd waarbij zijn testkennis, en inzicht in teststrategieën en testverbeteringen aangesproken werden, onder andere binnen de Rotterdamse haven. Hij heeft ook praktische ervaring als lijnmanager van een testafdeling, onder meer bij KPN, waar hij verantwoordelijk was voor het opzetten van de testafdeling en de operationele testwerkzaamheden.

Binnen Collis is Derk-Jan verantwoordelijk voor het begeleiden en coachen van de medewerkers. Hierbij hoort het inhoudelijk ondersteunen van de medior- en seniortesters, alsmede het opleiden van nieuwe medewerkers. Dit laatste doet hij onder meer door het verzorgen van de trainee klassen, waarin trainees gericht opgeleid worden om te kunnen functioneren als professioneel tester.

Daarnaast deelt hij zijn kennis graag met (potentiële) testcollega's. Hij onderhoudt contacten met verschillende onderwijsinstellingen en verzorgd regelmatig gastcollege's. Als docent en als spreker op internationale congressen staat Derk-Jan bekend als een gepassioneerd spreker, die goed in staat is een link te leggen tussen theorie en praktijk.