

# TESTAUTOMATISERING WENDBAAR ORGANISEREN

KLAAR VOOR DE TOEKOMST



Jos van Rooyen, Danny Greefhorst, Marcel Mersie

# TESTAUTOMATISERING WENDBAAR ORGANISEREN

## Andere uitgaven bij Van Haren Publishing

Van Haren Publishing (VHP) is gespecialiseerd in uitgaven over Best Practices, methodes en standaarden op het gebied van de volgende domeinen:

- IT en IT-management;
- Enterprise-architectuur;
- Projectmanagement, en
- Businessmanagement.

Deze uitgaven zijn beschikbaar in meerdere talen en maken deel uit van toonaangevende series, zoals *Best Practice*, *The Open Group series*, *Project management* en *PM series*.

Van Haren Publishing is tevens de uitgever voor toonaangevende instellingen en bedrijven, onder andere: Agile Consortium, ASL BiSL Foundation, CA, Centre Henri Tudor, CM Partners, Gaming Works, IACCM, IAOP, IPMA-NL, ITSqc, NAF, KNVI, PMI-NL, PON, The Open Group, The SOX Institute.

Onderwerpen per domein zijn:

### IT and IT Management

ABC of ICT  
ASL®  
CMMI®  
COBIT®  
e-CF  
ISM  
ISO/IEC 20000  
ISO/IEC 27001/27002  
ISPL  
IT4IT®  
IT-CMF™  
IT Service CMM  
ITIL®  
MOF  
MSF  
SABSA  
SAF  
SIAM™  
TRIM  
VersiSM™

### Enterprise Architecture

ArchiMate®  
BIAN  
GEA®  
Novius Architectuur Methode  
TOGAF®

### Business Management

*BABOK® Guide*  
BiSL® and BiSL® Next  
BRMBOK™  
BTF  
CATS CM®  
EFQM  
eSCM  
FSM  
IACCM  
ISA-95  
ISO 9000/9001  
OBM  
OPBOK  
SixSigma  
SOX  
SqEME®

### Project Management

A4-Projectmanagement  
DSDM/Atern  
ICB / NCB  
ISO 21500  
MINCE®  
M\_o\_R®  
MSP®  
P3O®  
*PMBOK® Guide*  
Praxis®  
PRINCE2®

Voor een compleet overzicht van alle uitgaven, ga naar onze website: [www.vanharen.net](http://www.vanharen.net)

# **Testautomatisering wendbaar organiseren**

Klaar voor de toekomst

Jos van Rooyen, Danny Greefhorst en Marcel Mersie



# Colofon

Titel: Testautomatisering wendbaar organiseren  
Ondertitel: Klaar voor de toekomst  
Auteurs: Jos van Rooyen, Danny Greefhorst en Marcel Mersie  
Uitgever: Van Haren Publishing, 's-Hertogenbosch, [www.vanharen.net](http://www.vanharen.net)

ISBN Hard copy: 978 94 018 0651 0  
ISBN eBook (pdf): 978 94 018 0652 7  
ISBN ePUB: 978 94 018 0653 4

Druk: Eerste druk, eerste oplage, mei 2020  
Lay-out en ontwerp: Coco Bookmedia, Amersfoort  
Copyright: © Van Haren Publishing, 2020

Voor verdere informatie over Van Haren Publishing, e-mail naar: [info@vanharen.net](mailto:info@vanharen.net).

Niets uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm, of op welke wijze ook, zonder voorafgaande schriftelijke toestemming van de uitgever.

No part of this publication may be reproduced in any form by print, photo print, microfilm or any other means without written permission by the publisher.

Hoewel deze uitgave met veel zorg is samengesteld, aanvaarden auteur(s) noch uitgever enige aansprakelijkheid voor schade ontstaan door eventuele fouten en/of onvolkomenheden in deze uitgave.

# Voorwoord

Het vakgebied van informatieprofessionals is enorm in beweging, en het is geweldig om te zien dat collega's en leden van de Koninklijke Vereniging van Informatieprofessionals (KNVI) daar ook mede richting aan geven. Met deze uitgave geven de auteurs een belangrijke bijdrage aan de 'body of knowledge' van vakgenoten. Ze onderzoeken een aantal belangrijke ontwikkelingen, beschrijven die én maken ze toepasbaar voor anderen. Dat doen ze vanuit kennis en kunde, en ervaring in het inzetten daarvan.

Om duiding te geven aan die veranderende wereld van het vakgebied én de wereld om ons heen, introduceerde de beroepsorganisatie de term 'Smart Humanity'. Daarmee wordt het vak van informatieprofessionals opnieuw gedefinieerd: Het gaat niet meer om alleen de inzet van vakkennis, maar om een veel bredere bijdrage aan de maatschappij. Met andere woorden, informatieprofessionals maken techniek en helpen bij het implementeren ervan. Vakgenoten zijn zowel makers als gebruikers als opleiders, en ze faciliteren andere mensen en organisaties in het verkrijgen van toegang tot informatie. Dat betekent dat er een belangrijke verantwoordelijkheid ligt bij informatieprofessionals. Ze bewaken het welzijn van de mensheid in relatie tot de inzet van techniek en de gevolgen van opeenvolgende disruptieve golven in de maatschappij.

Met dit boek laten de auteurs zien op welke wijze een aantal van deze belangrijke ontwikkelingen samenkomen. Ten eerste wordt ingegaan op de relatie tussen testen en architectuur. Daarmee leggen ze nog nadrukkelijker dan voorheen de relatie tussen deze vakgebieden. Vanuit iedere discipline vergt dit zowel diepe kennis van het vak, als de intentie om vanuit deze expertise in gesprek te zijn, samen te werken en samen te bouwen aan het nóg beter maken van informatiesystemen.

Ten tweede leggen ze de relatie tussen mensen, processen en de organisatie. Een drieluik van onderwerpen die helaas nog te weinig in samenhang worden beschouwd bij verbeteringen en vernieuwingen. Nieuwe methoden en technieken leveren vaak een verschuiving van taken en rollen op, en een verandering in gevraagde competenties. In dit boek wordt aangegeven wat de impact is van de inzet van testautomatisering. Het Europese Competence Framework (e-CF) waar ook de KNVI voorstander van is, kan helpen bij de verdere duiding van deze competenties.

Ten derde worden ook volwassenheid en veranderpotentieel geduid in dit boek. Nieuwe technieken hebben niet alleen impact op de manier waarop organisaties werken, maar moeten ook landen. En dat kost tijd en energie – een logische constatering waarvoor in de praktijk niet altijd oog is. Veranderen is niet alleen veranderkundig een opgave, maar levert vaak ook diverse dilemma's op. Dilemma's in bijvoorbeeld organisatie-inrichting of in de positie van mensen binnen organisaties. Soms levert het zelfs ethische vraagstukken op. Vanuit de KNVI wordt daarvoor onder meer aangesloten bij de ethische *code of conduct* van de internationale koepelorganisatie, de International Federation for Information Processing (IFIP),

om het gesprek te kunnen voeren over dergelijke dilemma's. Met een groeimodel geven de auteurs in dit boek hulp voor dergelijke keuzes die organisaties moeten maken bij het organiseren van testautomatisering.

We danken de auteurs voor deze bijdrage aan het vak, en we wensen u als lezer een inspirerende, vakinhoudelijke en mooie leestocht toe door dit boek.

Wouter Bronsgeest & Paul Baak  
duo-voorzitters KNVI

# Inhoud

<b>1</b>	<b>Inleiding</b>	<b>1</b>
1.1	Aanleiding	1
1.2	Structuur van het boek	4
<b>2</b>	<b>Wat en waarom testautomatisering?</b>	<b>7</b>
2.1	Definitie	7
2.2	De impact en valkuilen van Agile en DevOps	8
2.3	Redenen voor het toepassen van testautomatisering	9
2.4	Toepassingsgebieden voor testautomatisering	10
2.5	De aanpak van testautomatisering	12
2.6	De randvoorwaarden	14
2.7	Samenvatting	15
<b>3</b>	<b>De architectuur van testautomatisering</b>	<b>17</b>
3.1	Inleiding	17
3.2	Mens	18
3.3	Organisatie	19
3.4	Proces	21
3.5	Data	23
3.6	Technologie	24
3.7	Samenvatting	27
<b>4</b>	<b>Een groeimodel voor testautomatisering</b>	<b>29</b>
4.1	Inleiding	29
4.2	Ontwikkelniveaus	30
4.3	Mens	31
4.4	Organisatie	32
4.5	Proces	33
4.6	Data	34
4.7	Technologie	34
4.8	Toepassing	35
4.9	Samenvatting	37
<b>5</b>	<b>De waarde van testautomatisering</b>	<b>39</b>
5.1	Inleiding	39
5.2	Elementen van invloed op de business case	39
5.3	Return on investment van testautomatisering	41
5.4	Samenvatting	43



<b>6</b>	<b>De impact van testautomatisering op mens, organisatie en proces</b>	<b>45</b>
6.1	Inleiding	45
6.2	De positionering in de organisatie	46
6.3	Verandering van rol	47
6.4	Verandering van ontwikkelpad	48
6.5	Teamsamenstelling	50
6.6	Processen voor testen en testautomatisering	50
6.7	Verandering van proces	52
6.8	Werkwijzen voor testautomatisering	54
6.9	Toekomstvaste inrichting	56
6.10	Samenvatting	57
<b>7</b>	<b>Testdatamanagement</b>	<b>59</b>
7.1	Inleiding	59
7.2	Kwaliteit van testdata	60
7.3	Privacy	61
7.4	Inrichting van testdatamanagement	62
7.5	Synthetische testdata of productiedata	63
7.6	Opvoeren testdata	64
7.7	Herstellen testdata	65
7.8	Samenvatting	66
<b>8</b>	<b>Technologie</b>	<b>67</b>
8.1	Inleiding	67
8.2	Het drielagen-model	67
8.3	Referentie-architectuur	70
8.4	Toolselectie	73
<b>9</b>	<b>Testautomatiseringsarchitectuur</b>	<b>75</b>
9.1	Inleiding	75
9.2	Doelstellingen en eisen	76
9.3	Mens en organisatie	76
9.4	Proces	77
9.5	Data	78
9.6	Technologie	79
<b>10</b>	<b>Case herbruikbaarheid</b>	<b>81</b>
10.1	Situatieschets	81
10.2	De aanpak	82
10.3	Ervaringen	83
10.4	Lessons learned	83
10.5	Resultaten	84

<b>11</b>	<b>Case overdraagbaarheid</b>	<b>85</b>
11.1	Situatieschets	85
11.2	De aanpak	86
11.3	Lessons learned	88
11.4	Resultaten	88
<b>12</b>	<b>Casebeschrijving herhaalbaarheid</b>	<b>89</b>
12.1	Situatieschets	89
12.2	De aanpak	89
12.3	Resultaten	92
<b>13</b>	<b>Dankwoord</b>	<b>93</b>
	<b>Bijlage A Literatuur</b>	<b>94</b>
	<b>Bijlage B Web-links</b>	<b>95</b>
	<b>Bijlage C Begrippenlijst</b>	<b>96</b>
	<b>Bijlage D Over de auteurs</b>	<b>98</b>
	<b>Bijlage E Over KNVI</b>	<b>99</b>
	<b>Index</b>	<b>101</b>



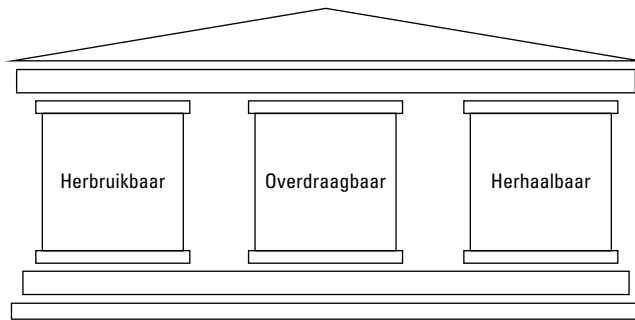
# 1 Inleiding

## 1.1 Aanleiding

Testautomatisering is geen nieuw onderwerp; er is al veel over gezegd en geschreven. Toch hebben wij gemeend om dit boek uit te brengen. Mensen en organisaties worden in toenemende mate afhankelijk van IT en dat vraagt ook expliciete aandacht voor de kwaliteit van software en de rol van testen daarbinnen. Testautomatisering blijkt ook in de context van Agile en DevOps een cruciaal instrument voor het verhogen van de kwaliteit van software. Door de toenemende snelheid in softwareontwikkeling is handmatig testen niet meer toereikend. Het tempo is simpelweg niet bij te houden. In een Agile/DevOps-context neemt de snelheid van ontwikkeling en releases toe. Continuous Integration en Continuous Delivery (CI/CD) zijn standaard aan het worden. Dat betekent dat ontwikkelaars hun wijzigingen frequent integreren en dat de resultaten automatisch in productie worden genomen. In zo'n context is het niet meer reëel om steeds alle testen handmatig uit te voeren. CI/CD vraagt om continuous testing, waarbinnen testautomatisering een centrale rol speelt. Testautomatisering in deze context gaat dan ook verder dan alleen de regressietest te automatiseren. Testautomatisering moet in de gehele ontwikkelketen worden toegepast. Dit staat bekend als de 'shift to the left'-beweging, wat inhoudt dat zo vroeg mogelijk getest moet worden en liefst ook geautomatiseerd.

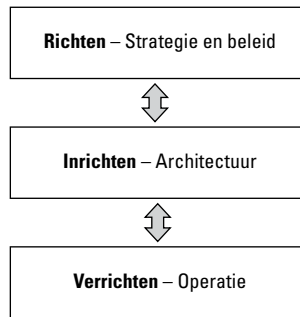
De ervaring leert dat binnen Agile teams direct gestart wordt met het toepassen van testautomatisering. Dat is ons inziens ook een absolute must. De kennis en opzet beperkt zich echter vaak tot een team of project en is moeilijk overdraagbaar naar andere teams of projecten. Er wordt vaak gekozen voor een te technische insteek, met vooral aandacht voor tools en te weinig aandacht voor mens, organisatie, data en proces. Hoewel deze insteek prima kan zijn voor een individueel team, is het moeilijk organisatiebreed op te schalen. De uitdaging is om testautomatisering op een toekomstvaste manier in te richten zodat ook anderen in de organisatie ervan kunnen profiteren. Wij zien juist op dit punt veel testautomatiseringsprojecten stranden. We denken dat expliciete aandacht nodig is voor het herbruikbaar, overdraagbaar en herhaalbaar maken van testautomatisering. Bij elkaar zijn dit drie belangrijke pijlers onder de inrichting van testautomatisering (zie ook figuur 1). Dit vraagt onder meer het expliciet vastleggen en verspreiden van kennis, maar heeft ook invloed op de manier waarop individuele testgevallen en testscripts worden opgesteld.

We denken verder dat een architectuurgebaseerde benadering noodzakelijk is. Door expliciete aandacht voor mens, organisatie, proces, data en technologie en hun samenhang is het mogelijk om testautomatisering organisatiebreed te verankeren en ook op lange termijn succesvol te laten zijn. Architectuur verbindt strategie en beleid aan de operatie. Het zorgt voor een vertaling van de richting naar een inrichting zodat vervolgens medewerkers hun werk goed kunnen verrichten. Het vestigt expliciete aandacht op het maken van inrichtingskeuzes. Het is een instrument om mee te organiseren. Architectuur gaat ook over het maken van modellen en dat heeft ook veel waarde in de context van testautomatisering om deze



Figuur 1 De drie pijlers onder testautomatisering

toekomstvast in te richten. Het vestigt meer aandacht op het belang van het specificeren van het te testen systeem. Dit maakt het testen en de testautomatisering eenvoudiger. In de ideale situatie is het gehele softwareontwikkelproces modelgebaseerd zodat architectuur, softwareontwikkeling en testen optimaal op elkaar aansluiten.



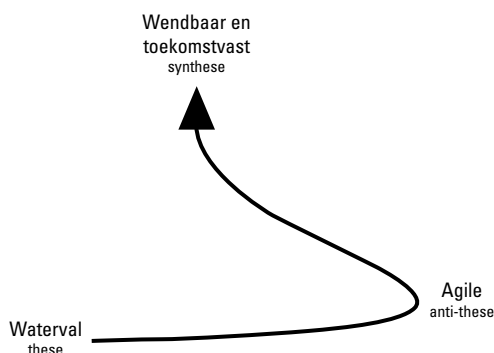
Figuur 2 Architectuur als verbindende sleutel

Dit boek gaat daarom uit van een architectuurgebaseerde benadering van testautomatisering. Het kijkt integraal naar alle relevante aspecten. Het geeft antwoord op vragen zoals: “Hoe borg je de opgedane kennis met betrekking tot testautomatisering in de gehele organisatie?” en “Hoe maak je testautomatisering overdraagbaar en herhaalbaar naar andere teams in de organisatie of in de keten?”. We reiken principes, richtlijnen en modellen aan over hoe testautomatisering toekomstvast ingericht kan worden. Daarbij zoeken we de expliciete aansluiting bij Agile en DevOps ontwikkelingen. We willen teams en organisaties helpen bij het maken van goede keuzes voor de inrichting van testautomatisering. Dat doen we door de kracht van testautomatisering te laten zien en de gevolgen van keuzes inzichtelijk te maken.

Het boek legt de nadruk op het belang van het organiseren van testautomatisering. Tools zijn een onderdeel van testautomatisering, maar de belangrijkste opgave ligt bij het ervoor zorgen dat mensen de juiste dingen gaan doen. Dat gaat veel meer over kennis, vaardigheden, overtuigingen en rolverdelingen dan dat het gaat over frameworks, technologie, tools of scripts. Wij denken dat testautomatisering een structurele inrichting vraagt en daarmee niet iets is dat je alleen aan individuele teams kunt laten. De kunst bestaat eruit om kennis, kunde en energie van mensen te bundelen en te hergebruiken en daarmee de organisatie als

geheel op een hoger plan te tillen. Dit vraagt een combinatie van top-down en bottom-up organiseren. Architectuur is daarbij een belangrijke sleutel. Het biedt een vorm van top-down structurering, maar laat teams tevens de ruimte om binnen hun eigen context de juiste keuzes te maken. Het geeft richting zonder te beknellen. De combinatie van de vakgebieden architectuur en testen maakt dit boek uniek.

Dit boek kun je zien als een volgende fase in de ontwikkeling van de visie op softwareontwikkeling en testen. Traditionele softwareontwikkeling volgde een watervalbenadering. Kenmerk daarvan is het inbouwen van zekerheden door momenten in te lassen waarop producten worden vastgezet, voordat een volgende fase wordt gestart. Deze zekerheden bleken schijnzekerheden. Niets blijkt zo veranderlijk als de eisen en wensen van gebruikers. De tegenbeweging is agile, dat er juist vanuit gaat dat softwareontwikkeling inherent moet omgaan met onzekerheden. Dat is op zich een goed inzicht, maar de valkuil is dat alles dat verder vooruitkijkt als onbelangrijk wordt gezien. Deze kortzichtigheid zorgt ervoor dat organisaties voor ongewenste verrassingen komen te staan. Voor testautomatisering leidt dat tot automatische tests die het op een gegeven moment gewoon niet meer doen of die niet buiten de context van een project of team bruikbaar blijken. De volgende fase is ons inziens een combinatie van ideeën uit de waterval- en de agile fase. We noemen deze fase “wendbaar en toekomstvast”. Het adopteert de flexibiliteit van de agile fase met de toekomstgerichtheid van de waterval fase.



Figuur 3 Ontwikkeling van visie op softwareontwikkeling en testen

De doelgroep van het boek is breder gedefinieerd dan de traditionele testcommunity. Het is ook geschreven voor IT-managers, IT-architecten, informatiemangers, adviseurs en mensen die beslissingen nemen over de inrichting van IT-gerelateerde veranderingen. Het boek kan testspecialisten helpen in het verkrijgen van een breder perspectief op relevante aspecten, waardoor ze in een ander licht worden geplaatst. Het probeert met name alternatieve kennis, ideeën en inzichten aan te reiken, waardoor je anders naar testautomatisering kunt kijken.

## 1.2 Structuur van het boek

Hoofdstuk 2 van dit boek gaat in op wat testautomatisering is en waarom het belangrijk is. Het zorgt voor een algemeen begrip van het onderwerp en beschrijft ook een aantal belangrijke randvoorwaarden om het te laten werken.

Hoofdstuk 3 beschrijft de architectuur van testautomatisering door vanuit verschillende aspecten te kijken naar het onderwerp en daarbinnen richtinggevende uitspraken te doen in de vorm van architectuurprincipes.

Hoofdstuk 4 verdiept de aspecten en de architectuurprincipes door in meer detail te beschrijven hoe organisaties kunnen groeien op het gebied van testautomatisering. Het bevat een groeimodel met verschillende volwassenheidsniveaus waarbinnen concrete vragen zijn geformuleerd.

Hoofdstuk 5 beschrijft elementen voor het bepalen van de kosten en baten van testautomatisering. Op basis van deze elementen kan waar nodig een business case worden opgesteld voor testautomatisering. Naast ondersteuning van besluitvorming kan het anderen ook inzichten geven in de overwegingen.

Hoofdstuk 6 beschrijft in meer detail wat de impact is van testautomatisering op mens, proces en organisatie. Het laat zien hoe de verschillende testrollen schuiven door de invoering van testautomatisering. Het geeft ook inzicht in de activiteiten die zullen veranderen.

Hoofdstuk 7 geeft inzicht in testdatamanagement. Het laat zien dat het sturen op de kwaliteit van testdata een belangrijke succesfactor is voor testautomatisering. Het schetst ook keuzes voor het omgaan met testdata zoals het gebruik van synthetische testdata of het anonimiseren van productiedata.

Hoofdstuk 8 beschrijft de impact van testautomatisering op technologie. Het biedt onder meer een referentie-architectuur die inzicht geeft in de functionaliteiten van testtools die gebruikt kunnen worden voor het selecteren van tools.

Hoofdstuk 9 stelt voor dat testautomatisering wordt ondersteund door een specifieke testautomatiseringsarchitectuur. Het biedt ook een standaard structuur voor het beschrijven van belangrijke keuzes voor de inrichting van testautomatisering, inclusief voorbeeldcontent.

Hoofdstukken 10, 11 en 12 bevatten beschrijvingen van cases die de hoofdboodschap van dit boek ondersteunen: het toekomstvast inrichten van testautomatisering door expliciete aandacht voor herbruikbaarheid, overdraagbaarheid en herhaalbaarheid. We hebben drie organisaties bereid gevonden om hun ervaringen met betrekking tot het toekomstvast inrichten van testautomatisering te beschrijven.

Dit boek legt een expliciete relatie tussen testautomatisering en architectuur. In hoofdstuk 3 beschrijven we architectuurprincipes voor testautomatisering. De consequenties van de principes zijn vertaald naar hoofdstukken over specifieke onderwerpen. Dat geldt met name voor hoofdstuk 4 waarin het groeimodel is beschreven; veel van wat daarin staat is een consequentie van de principes in dit hoofdstuk over architectuur. We hanteren in een aantal hoofdstukken ook de indeling in de vijf aspecten die we in dit hoofdstuk hanteren: mens, organisatie, proces, data en technologie. We presenteren in hoofdstuk 8 een referentie-architectuur voor testtools, waarmee deze kunnen worden gepositioneerd en geselecteerd. Tenslotte presenteren we in hoofdstuk 9 een sjabloon voor een testautomatiseringsarchitectuur. Dat is een architectuurdocument (of onderdeel van een ander architectuurdocument) dat specifiek ingaat op testautomatisering.





# 2 Wat en waarom testautomatisering?

## 2.1 Definitie

Wij definiëren testautomatisering als het gebruik van software om testactiviteiten uit te voeren of te ondersteunen. Dat kan bijvoorbeeld gaan over testbeheer, testontwerp of testuitvoering. Testautomatisering neemt een aantal werkzaamheden over van het testteam. Denk daarbij aan het managen van een testproject, het ontwikkelen van testgevallen, het uitvoeren van testgevallen en het managen van de testresultaten uitgedrukt in bevindingen of testresultaten. In dit boek beschouwen we testautomatisering in de breedste zin van het woord waarbij de gehele businessketen wordt ondersteund. Testautomatisering wordt al zeer lang ingezet, maar met wisselend succes. De ontwikkeling van testautomatisering kun je indelen naar diverse generaties van ontwikkeling. Inmiddels zijn we met de vierde generatie bezig.

De eerste generatie testautomatisering kenmerkte zich door het opnemen en vervolgens afspelen van testgevallen. Een krachtig mechanisme maar met een grote tekortkoming m.b.t. de herhaalbaarheid van de testgevallen. Het onderhouden van de geautomatiseerde testgevallen vroeg relatief veel inspanning doordat elke wijziging in de gebruikersinterface leidde tot wijzigingen in de testgevallen. Daarnaast moet het te testen systeem eigenlijk al klaar zijn en een mate van stabiliteit hebben bereikt voordat gestart kan worden met het definiëren van testgevallen. De tweede generatie kwam deels tegemoet aan de bezwaren van herhaalbaarheid. Het opnemen en afspelen werd aangepast door functies te programmeren waarbij de benodigde invoer voor de test werd gespecificeerd als variabelen die eenvoudig aangepast konden worden. Dit betekende een verbetering ten opzichte van de eerste generatie maar het was nog niet ideaal. In de derde generatie werd de testautomatisering meer data-gestuurd ingericht. Een werkelijk grote stap voorwaarts waarbij de toepassing van testtools eenvoudiger werd door de goede gebruikersinterface en het uitvoeren van functies door de testtools. Activiteiten van de gebruiker werden door het testtool overgenomen. Denk bijvoorbeeld aan het genereren van testdata, opvoeren van testscripts en het rapporteren van de resultaten. Ondanks deze ontwikkelingen blijkt het lastig te zijn om testautomatisering daadwerkelijk toekomstvast in te richten. We bedoelen daarmee het herhaalbaar, overdraagbaar en herbruikbaar inrichten van testautomatisering waardoor tests continu kunnen worden uitgevoerd en overgedragen zonder een al te grote inspanning. Het toekomstvast inrichten van testautomatisering noemen we de vierde generatie testautomatisering en dat is het uitgangspunt voor dit boek.

De volgende paragraaf beschrijft de impact van Agile en DevOps op testen en testautomatisering. Vervolgens wordt ingegaan op belangrijke redenen voor het toepassen van testautomatisering. Daarna gaat dit hoofdstuk dieper in op wat testautomatisering is, wat in welke situaties het best kan worden toegepast en welke randvoorwaarden moeten zijn ingevuld om de kans op succes zo groot mogelijk te maken.

## 2.2 De impact en valkuilen van Agile en DevOps

De ontwikkeling van de vierde generatie testautomatisering is door de aandacht voor Agile, DevOps en CI/CD in een stroomversnelling terecht gekomen. Agile softwareontwikkeling kenmerkt zich door het snel en kort-cyclisch opleveren van een werkend product (bijvoorbeeld iedere twee weken). Testen moet mee kunnen in dit tempo. Daardoor wordt een andere benadering gevraagd om de kwaliteit van de software aan te tonen. Testautomatisering is meer een uitgangspunt geworden dan dat de waarde ervan vooraf moet worden aangetoond. De rol van testen en testautomatisering is door Agile softwareontwikkeling ook meer fundamenteel aan het veranderen (zie Tabel 1). Het is toenemend geïntegreerd in het gehele softwareontwikkelproces, waarbij testen komt vóór programmeren, een meer continue activiteit is geworden en niet meer los is georganiseerd. Het is compleet verankerd in de ontwikkelketen. Door de aandacht voor DevOps wordt ook de levenscyclus van de tests belangrijker en moet je verder kijken dan de initiële ontwikkeling. Dit alles vraagt om een toekomstvaste inrichting van testautomatisering.

Tabel 1 Fundamentele veranderingen in testen door Agile en DevOps

Van	Naar
Testautomatisering als er een business case is	Testautomatisering is uitgangspunt
Testen achteraf maken	Eerst test maken, dan pas code maken
Testen per release	Continu testen
Apart testteam	Testers als onderdeel van ontwikkelteam
Tester als rol	Testen als competentie
Testen voor initiële ontwikkeling	Testen voor ontwikkeling en beheer

Het blijkt geen sinecure te zijn om testautomatisering toekomstvast in te richten. Projecten en organisaties beginnen enthousiast maar na verloop van tijd zie je het enthousiasme afnemen. Daar zijn allerlei redenen voor. De belangrijkste reden is dat het vaak als een éénmalige inspanning wordt gezien en alleen wordt uitgevoerd in de context van een project. Na verloop van tijd is het project gereed, maar is het beheer van de resultaten niet goed geregeld. Er wordt te weinig tijd gereserveerd om testscripts te onderhouden. Het resultaat van alle inspanningen gaat dan snel verloren. Het herhalen en hergebruiken van tests wordt steeds lastiger. De testautomatisering kan de frequentie van opleveren van software niet meer bijhouden. Het van tevoren nadenken over de opzet van testautomatisering en het borgen van de projectresultaten is daarom essentieel. Een toekomstvaste architectuur vormt daarbij een belangrijke basis.

Inherent zorgt Agile softwareontwikkeling ook voor een aantal uitdagingen. Agile wordt al afgezet tegen traditionele softwareontwikkeling, waarbij alle eerdere werkwijzen als achterhaald wordt gezien. Typische valkuilen daarbij zijn onder meer:

- Denken dat teams alles zelf kunnen bedenken;
- Denken dat alles draait om functionaliteit;
- Denken dat documentatie niet meer nodig is;
- Denken dat alle teamleden ontwikkelaar zijn.