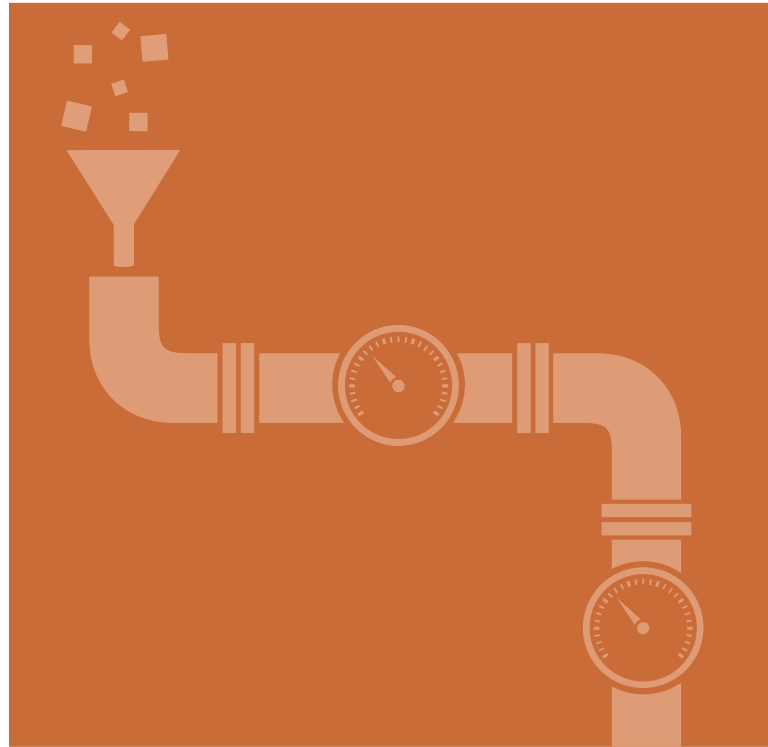
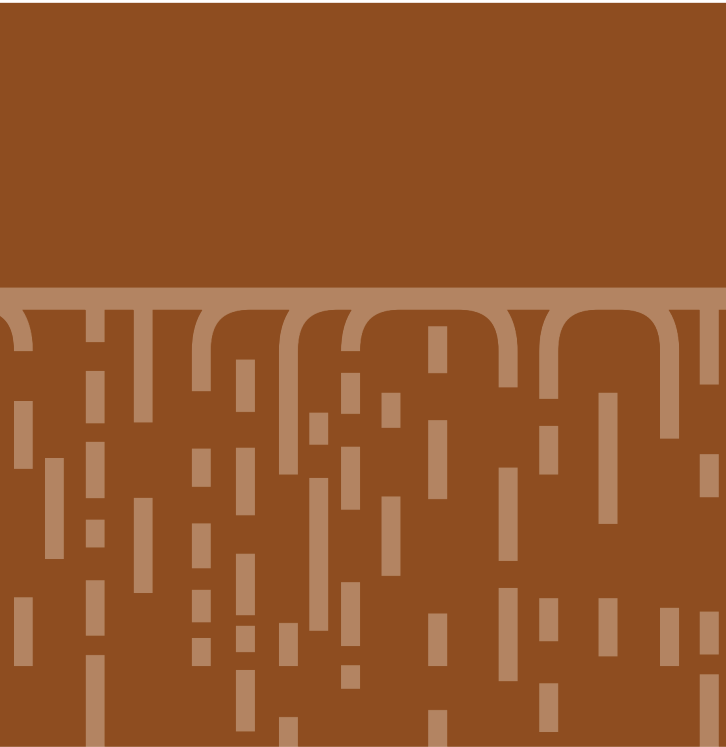


# CONTINUOUS DESIGN

Een publicatie in de **Continuous Everything** reeks



BART DE BEST



# **DevOps Continuous Design Best Practices**

Een uitgave in de Continuous Everything reeks

Bart de Best

Onder redactie van  
Louis van Hemmen

# Colofon

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

Leonon Media  
(0)572 - 851 104

Algemene vragen : info@leonon.nl  
Sales vragen : verkoop@leonon.nl  
Manuscript / auteur : redactie@leonon.nl

© 2022 Leonon Media

Omslagontwerp : Eric Coenders, IanusWeb, Nijmegen  
Productie : Printforce B.V., Culemborg

Titel : DevOps Continuous Design  
Sub titel : Een uitgave in de Continuous Everything reeks  
Datum : 3 december 2022  
Auteur : Bart de Best  
Uitgever : Leonon Media  
ISBN13 : 978 94 92618 481  
Druk : Eerste druk, zevende editie, 3 december 2022

© 2022, Leonon Media

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

## TRADEMARK NOTICES

ArchiMate® and TOGAF® are registered trademarks of The Open Group.  
COBIT® is a registered trademark of the Information Systems Audit and Control Association (ISACA) / IT Governance Institute (ITGI).  
ITIL® and PRINCE2® are registered trademarks of Axelos Limited.  
Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc.

***"We build our computer (systems)  
the way we build our cities:  
over time, without a plan, on top of ruins."***

by Ellen Ullma

# Inhoudsopgave

<b>1</b>	<b>INTRODUCTIE .....</b>	<b>1</b>
1.1	DOEL .....	1
1.2	DOELGROEP .....	1
1.3	ACHTERGROND.....	1
1.4	STRUCTUUR .....	4
1.4.1	HOOFDSTUK 2: BASISCONCEPTEN EN BASISBEGRIPPEN .....	4
1.4.2	HOOFDSTUK 3: CONTINUOUS DESIGN DEFINITIE .....	4
1.4.3	HOOFDSTUK 4: CONTINUOUS DESIGN VERANKERING .....	4
1.4.4	HOOFDSTUK 5: CONTINUOUS DESIGN ARCHITECTUUR .....	4
1.4.5	HOOFDSTUK 6: CONTINUOUS DESIGN ONTWERP .....	4
1.4.6	HOOFDSTUK 7: BUSINESS VIEW .....	4
1.4.7	HOOFDSTUK 8: SOLUTION VIEW .....	4
1.4.8	HOOFDSTUK 9: DESIGN VIEW .....	4
1.4.9	HOOFDSTUK 10: REQUIREMENT VIEW .....	4
1.4.10	HOOFDSTUK 11: TEST VIEW .....	4
1.4.11	HOOFDSTUK 12: CODE VIEW .....	4
1.4.12	HOOFDSTUK 13: CONTINUOUS DESIGN BIJ ASSURITAS.....	5
1.4.13	HOOFDSTUK 14: CONTINUOUS DESIGN ASSESSMENT .....	5
1.5	BIJLAGEN.....	5
1.6	LEESWIJZER.....	5
<b>2</b>	<b>BASISCONCEPTEN EN BASISBEGRIPPEN .....</b>	<b>7</b>
2.1	BASISCONCEPTEN.....	7
2.1.1	CONTINUOUS DESIGN PYRAMID .....	7
2.1.2	NON CONTINUOUS DESIGN PYRAMID.....	7
2.2	BASISBEGRIPPEN .....	8
2.2.1	BUSINESS VIEW .....	8
2.2.2	SOLUTION VIEW .....	8
2.2.3	DESIGN VIEW .....	9
2.2.4	REQUIREMENT VIEW .....	9
2.2.5	TEST VIEW.....	9
2.2.6	CODE VIEW .....	9
<b>3</b>	<b>CONTINUOUS DESIGN DEFINITIE.....</b>	<b>11</b>
3.1	ACHTERGROND.....	11
3.2	DEFINITIE .....	11
3.3	TOEPASSING.....	11
3.3.1	OP TE LOSSEN PROBLEMEN.....	11
3.3.2	DE ROOT CAUSE .....	12
<b>4</b>	<b>CONTINUOUS DESIGN VERANKERING .....</b>	<b>13</b>
4.1	HET VERANDERPARADIGMA .....	13
4.2	BEELDVORMING.....	14
4.2.1	WAT WILLEN WE? .....	14
4.2.2	WAT WILLEN WE NIET?.....	15
4.3	MACHTSVERHOUDING .....	15
4.3.1	WAT WILLEN WE? .....	16
4.3.2	WAT WILLEN WE NIET?.....	18
4.4	ORGANISATIEVORMGEVING .....	18
4.4.1	WAT WILLEN WE? .....	19
4.4.2	WAT WILLEN WE NIET?.....	19

4.5	RESOURCES .....	20
4.5.1	WAT WILLEN WE? .....	20
4.5.2	WAT WILLEN WE NIET? .....	20
4.6	WATERVAL VERSUS CONTINUOUS DESIGN .....	21
4.6.1	BEELD WATERVAL AANPAK .....	22
4.6.2	MACHT WATERVAL AANPAK .....	23
4.6.3	ORGANISATIE WATERVAL AANPAK .....	23
4.6.4	RESOURCES WATERVAL AANPAK .....	23
<b>5</b>	<b>CONTINUOUS DESIGN ARCHITECTUUR.....</b>	<b>25</b>
5.1	ARCHITECTUURPRINCIPES.....	25
5.1.1	ALGEMEEN .....	25
5.1.2	PEOPLE .....	25
5.1.3	PROCESS.....	25
5.1.4	TECHNOLOGY .....	27
5.2	ARCHITECTUURMODELLEN .....	27
5.2.1	CONTINUOUS DESIGN PYRAMID MODEL.....	27
5.2.2	CONTINUOUS DESIGN PLANNING MODEL .....	28
<b>6</b>	<b>CONTINUOUS DESIGN ONTWERP.....</b>	<b>29</b>
6.1	CONTINUOUS DESIGN VALUE STREAM .....	29
6.2	CONTINUOUS DESIGN USE CASE DIAGRAM .....	29
6.3	CONTINUOUS DESIGN USE CASE .....	30
<b>7</b>	<b>BUSINESS VIEW .....</b>	<b>35</b>
7.1	INLEIDING .....	35
7.2	SYSTEM CONTEXT DIAGRAM.....	35
7.2.1	DOELSTELLING.....	35
7.2.2	CONTENT.....	36
7.2.3	TE BEANTWOORDEN VRAGEN .....	36
7.2.4	TEMPLATE .....	36
7.2.5	TIPS EN TRUCS.....	36
7.2.6	VOORBEELD .....	37
7.3	VALUE STREAM CANVAS .....	38
7.3.1	DOELSTELLING .....	38
7.3.2	CONTENT.....	38
7.3.3	TE BEANTWOORDEN VRAGEN .....	38
7.3.4	TEMPLATE .....	38
7.3.5	TIPS EN TRUCS.....	40
7.3.6	VOORBEELD .....	40
<b>8</b>	<b>SOLUTION VIEW .....</b>	<b>43</b>
8.1	INLEIDING .....	43
8.2	USE CASE DIAGRAM.....	43
8.2.1	DOELSTELLING.....	43
8.2.2	CONTENT.....	44
8.2.3	TE BEANTWOORDEN VRAGEN .....	44
8.2.4	BASIS TEMPLATE.....	44
8.2.5	TEMPLATE MET INCLUDE USE CASE .....	44
8.2.6	TEMPLATE MET EXTEND USE CASE .....	45
8.2.7	TIPS EN TRUCS.....	45
8.2.8	VOORBEELD .....	46
8.3	SYSTEM BUILDING BLOCKS .....	47

8.3.1	DOELSTELLING .....	47
8.3.2	CONTENT .....	47
8.3.3	TE BEANTWOORDEN VRAGEN.....	47
8.3.4	TEMPLATE SYSTEM BUILDING BLOCKS – ALGEMEEN .....	48
8.3.5	TIPS EN TRUCS ALGEMEEN .....	48
8.3.6	TEMPLATE SYSTEM BUILDING BLOCKS – INFORMATION.....	49
8.3.7	TIPS EN TRUCS SBB-I .....	51
8.3.8	TEMPLATE SYSTEM BUILDING BLOCKS – APPLICATION .....	52
8.3.9	TIPS EN TRUCS SBB-A.....	53
8.3.10	TEMPLATE SYSTEM BUILDING BLOCKS – TECHNOLOGY.....	54
8.3.11	TIPS EN TRUCS SBB-T .....	56
8.3.12	VOORBEELD SBB-I, SBB-A EN SBB-T.....	56
8.4	VALUE STREAM MAPPING .....	58
8.4.1	DOELSTELLING .....	58
8.4.2	CONTENT .....	58
8.4.3	TE BEANTWOORDEN VRAGEN.....	59
8.4.4	TEMPLATE.....	59
8.4.5	TIPS EN TRUCS .....	59
8.4.6	VOORBEELD.....	59
<b>9</b>	<b>DESIGN VIEW.....</b>	<b>61</b>
9.1	INLEIDING .....	61
9.2	USE CASE .....	62
9.2.1	DOELSTELLING .....	62
9.2.2	CONTENT .....	62
9.2.3	TE BEANTWOORDEN VRAGEN.....	63
9.2.4	TEMPLATE USE CASE NARRATIVE.....	63
9.2.5	TIPS EN TRUCS USE CASE NARRATIVE.....	64
9.2.6	VOORBEELD USE CASE NARRATIVE .....	65
9.2.7	TEMPLATE USE CASE SCENARIO .....	67
9.2.8	TIPS EN TRUCS USE CASE SCENARIO .....	68
9.2.9	VOORBEELD USE CASE SCENARIO .....	68
<b>10</b>	<b>REQUIREMENT VIEW .....</b>	<b>69</b>
10.1	INLEIDING.....	69
10.1.1	BEHAVIOR DRIVEN DEVELOPMENT .....	69
10.1.2	WAT IS BDD? .....	70
10.1.3	WAT IS GHERKIN? .....	70
10.2	BEHAVIOR DRIVEN DEVELOPMENT.....	70
10.2.1	DOELSTELLING .....	70
10.2.2	CONTENT.....	70
10.2.3	TE BEANTWOORDEN VRAGEN .....	71
10.2.4	TEMPLATE.....	71
10.2.5	TIPS EN TRUCS .....	71
10.2.6	VOORBEELD .....	71
<b>11</b>	<b>TEST VIEW .....</b>	<b>73</b>
11.1	INLEIDING.....	73
11.2	TEST DRIVEN DEVELOPMENT .....	74
11.2.1	DOELSTELLING .....	74
11.2.2	CONTENT.....	74
11.2.3	TE BEANTWOORDEN VRAGEN .....	74
11.2.4	TEMPLATE.....	74

11.2.5	TIPS EN TRUCS .....	75
11.2.6	VOORBEELD .....	75
<b>12</b>	<b>CODE VIEW .....</b>	<b>77</b>
12.1	INLEIDING .....	77
12.2	CONTINUOUS DOCUMENTATION DEVELOPMENT .....	78
12.2.1	DOELSTELLING.....	78
12.2.2	CONTENT .....	78
12.2.3	TE BEANTWOORDEN VRAGEN .....	78
12.2.4	TEMPLATE .....	79
12.2.5	TIPS EN TRUCS .....	80
12.2.6	VOORBEELD .....	80
<b>13</b>	<b>CONTINUOUS DESIGN BIJ ASSURITAS .....</b>	<b>85</b>
13.1	ASSURITAS .....	85
13.1.1	DE ORGANISATIE.....	85
13.1.2	DE VISIE .....	85
13.1.3	DE DOELSTELLING .....	85
13.1.4	DE STRATEGIE .....	85
13.1.5	DE IST SITUATIE .....	85
13.1.6	DE SOLL SITUATIE.....	86
13.1.7	HET MIGRATIEPAD .....	86
13.2	CONTINUOUS DESIGN .....	87
13.2.1	DE VRAGEN.....	87
13.2.2	FSA BEELDVORMING .....	87
13.2.3	FSA MACHT .....	89
13.2.4	FSA ORGANISATIE.....	89
13.2.5	FSA RESOURCES .....	92
<b>14</b>	<b>CONTINUOUS DESIGN ASSESSMENT .....</b>	<b>93</b>
14.1	WAT IS HET CE-MODEL .....	93
14.2	VOLWASSENHEIDSDIMENSIES .....	96
14.3	DEVOPS CE MODEL, CN.....	97
	<b>BIJLAGE A, LITERATUURLIJST .....</b>	<b>103</b>
	<b>BIJLAGE B, BEGRIPPENLIJST .....</b>	<b>107</b>
	<b>BIJLAGE C, AFKORTINGEN .....</b>	<b>123</b>
	<b>BIJLAGE D, WEBSITES.....</b>	<b>127</b>
	<b>BIJLAGE E, INDEX .....</b>	<b>129</b>



## Figuren

FIGUUR 1-1, DEVOPS LEMNISCAAT. ....	1
FIGUUR 1-2, SoR, SoE EN SoI (SOURCE HSO THE RESULT COMPANY). ....	3
FIGUUR 2-1, CONTINUOUS DESIGN PYRAMID. ....	7
FIGUUR 2-2, NON CONTINUOUS DESIGN PYRAMID. ....	8
FIGUUR 4-1, VERANDERPARADIGMA. ....	13
FIGUUR 4-2, VERANDERPARADIGMA - BEELDVORMING. ....	14
FIGUUR 4-3, VERANDERPARADIGMA - MACHTSVERHOUDING. ....	16
FIGUUR 4-4, VERANDERPARADIGMA - ORGANISATIE. ....	19
FIGUUR 4-5, VERANDERPARADIGMA - RESOURCES. ....	20
FIGUUR 4-6, INGEVULD VERANDERPARADIGMA VOOR CONTINUOUS DESIGN KARAKTERISTIEKEN. ....	22
FIGUUR 4-7, INGEVULD VERANDERPARADIGMA VOOR WATERVAL DESIGN KARAKTERISTIEKEN. ....	22
FIGUUR 5-1, CONTINUOUS DESIGN PYRAMID MET DELIVERABLES EN TE BEANTWOORDEN VRAGEN. ....	27
FIGUUR 5-2, CONTINUOUS DESIGN PLANNING MODEL. ....	28
FIGUUR 6-1, CONTINUOUS DESIGN VALUE STREAM. ....	29
FIGUUR 6-2, USE CASE DIAGRAM VOOR CONTINUOUS DESIGN. ....	30
FIGUUR 7-1, BUSINESS VIEW. ....	35
FIGUUR 7-2, SYSTEM CONTEXT DIAGRAM TEMPLATE. ....	36
FIGUUR 7-3, SYSTEM CONTEXT DIAGRAM VOORBEELD. ....	37
FIGUUR 7-4, VALUE STREAM CANVAS TEMPLATE. ....	39
FIGUUR 7-5, VALUE STREAM CANVAS VOORBEELD. ....	41
FIGUUR 8-1, SOLUTION VIEW. ....	43
FIGUUR 8-2, USE CASE DIAGRAM BASIS TEMPLATE. ....	44
FIGUUR 8-3, USE CASE DIAGRAM TEMPLATE MET EEN INCLUDE USE CASE. ....	45
FIGUUR 8-4, USE CASE DIAGRAM TEMPLATE MET EXTEND USE CASE. ....	45
FIGUUR 8-5, 16-VLAKSMODEL. ....	46
FIGUUR 8-6, VOORBEELD USE CASE DIAGRAM 'KOFFIESERVICE'. ....	47
FIGUUR 8-7, TEMPLATE SYSTEM BUILDING BLOCKS – ALGEMEEN. ....	48
FIGUUR 8-8, TEMPLATE SYSTEM BUILDING BLOCKS – INFORMATIE. ....	50
FIGUUR 8-9, TEMPLATE SYSTEM BUILDING BLOCKS – APPLICATIE. ....	52
FIGUUR 8-10, TEMPLATE SYSTEM BUILDING BLOCKS – TECHNOLOGIE. ....	55
FIGUUR 8-11, VOORBEELD VAN INFORMATION SYSTEM BUILDING BLOCKS VAN EEN KOFFIESERVICE. ....	57
FIGUUR 8-12, VOORBEELD VAN APPLICATION SYSTEM BUILDING BLOCKS VAN EEN KOFFIESERVICE. ....	57
FIGUUR 8-13, VOORBEELD VAN TECHNOLOGY SYSTEM BUILDING BLOCKS VAN EEN KOFFIESERVICE. ....	58
FIGUUR 8-14, VALUE STREAM MAPPING TEMPLATE. ....	59
FIGUUR 8-15, VOORBEELD-1 VALUE STREAM MAPPING KOFFIESERVICE. ....	60
FIGUUR 8-16, VOORBEELD-2 VALUE STREAM MAPPING KOFFIESERVICE. ....	60
FIGUUR 9-1, DESIGN VIEW. ....	61
FIGUUR 9-2, TEMPLATE USE CASE SCENARIO. ....	67
FIGUUR 9-3, VOORBEELD VAN TWEE USE CASE SCENARIO'S. ....	68
FIGUUR 10-1, REQUIREMENT VIEW. ....	69
FIGUUR 10-2, BDD TEMPLATE. ....	71
FIGUUR 10-3, GHERKIN FEATUREFILE VOORBEELD. ....	72
FIGUUR 11-1, TEST VIEW. ....	73
FIGUUR 12-1, CODE VIEW. ....	77
FIGUUR 13-1, BEMORE VOOR EEN FSA BIJ ASSURITAS. ....	87
FIGUUR 13-2, FSA BEELDVORMING BIJ ASSURITAS. ....	88
FIGUUR 13-3, DE HUIDIGE EN DE GEWENSTE DOCUMENTATIE WERKWIJZE BIJ ASSURITAS. ....	88
FIGUUR 13-4, FSA MACHT BIJ ASSURITAS. ....	89
FIGUUR 13-5, FSA ORGANISATIE BIJ ASSURITAS. ....	90
FIGUUR 13-6, IMPLEMENTATION CONTINUOUS DESIGN AT ASSURITAS. ....	91

FIGUUR 13-7, FSA TEMPLATE.....	91
FIGUUR 13-8, FSA USAGE IN THE DEVELOPMENT PROCESS.....	92
FIGUUR 13-9, FSA RESOURCES.....	92
FIGUUR 14-1, DEVOPS CE-SPIDER MODEL.....	96
FIGUUR 14-2, DEVOPS CN-SPIDER MODEL.....	99

## Tabellen

TABEL 1-1, CONTINUOUS EVERYTHING ASPECTEN.....	2
TABEL 1-2, BIJLAGEN.....	5
TABEL 3-1, VEEL VOORKOMENDE PROBLEMEN BIJ HET HANTEREN VAN EEN DESIGN.....	12
TABEL 4-1, VERSCHILLEN TUSSEN EEN WATERVAL DESIGN EN EEN CONTINUOUS DESIGN.....	21
TABEL 6-1, USE CASE TEMPLATE.....	31
TABEL 6-2, USE CASE VOOR CONTINUOUS DESIGN.....	34
TABEL 9-1, USE CASE NARRATIVE TEMPLATE.....	64
TABEL 9-2, VOORBEELD VAN EEN USE CASE NARRATIVE.....	67
TABEL 11-1, PYTHON UNITTEST TEMPLATE.....	75
TABEL 11-2, PYTHON UNITTEST VOORBEELD.....	75
TABEL 12-1, PYTHON HEADER ANNOTATIE VOORBEELD.....	82
TABEL 12-2, C++ QT STYLE SCRIPT ANNOTATIEVOORBEELD.....	83
TABEL 12-3, APPLICATIEDOCUMENTATIE GEGENEREERD DOOR DOXYGEN.....	84
TABEL 14-1, DEVOPS CE-MODEL.....	93
TABEL 14-2, CONTINUOUS EVERYTHING.....	94
TABEL 14-3, CMMI LEVELS VOOR CONTINUOUS EVERYTHING.....	95
TABEL 14-4, PR-ORG-009. VOLWASSENHEIDSNIVEAUS.....	96
TABEL 14-5, CN MATURITY CHARACTERISTICS.....	98

## Bijlagen

BIJLAGE A, LITERATUURLIJST.....	103
BIJLAGE B, BEGRIPPENLIJST.....	107
BIJLAGE C, AFKORTINGEN.....	123
BIJLAGE D, WEBSITES.....	127
BIJLAGE E, INDEX.....	129

## Voorwoord

Dit boek is samengesteld op basis van mijn ervaringen bij het ontwerpen van informatiesystemen in een DevOps omgeving. Het is een momentopname van de best practices die ik nu hanteer. Gezien de snelheid waarmee de wereld van DevOps zich ontwikkelt en de behoefte om u met zo min mogelijk tekst zo veel mogelijk beelden te geven van het hanteren van een Continuous Design, heb ik besloten dit boek Agile te houden. Dit houdt in dat het zeer beknopt beschrijft wat belangrijke inzichten zijn, die ik heb opgedaan gedurende mijn rol als consultant, trainer, coach en examiner ten aanzien van Continuous Design gerelateerd werk. Waar van toepassing verwijs ik naar bronnen die ik zelf heb geraadpleegd om mij verder te bekwamen. Hierbij besef ik dat deze best practices niet voor alle informatiesystemen van toepassing zullen zijn en dat de aanpak een momentopname is die wellicht door de toenemende snelheid van innovatie weer achterhaald kan zijn.

Veel van mijn ervaringen heb ik ook al gedeeld in de artikelen op [www.ITpedia.nl](http://www.ITpedia.nl). Tevens heb ik de kennis en kunde vertaald naar diverse trainingen die ik verzorg. Deze zijn te vinden op [www.dbmetrics.nl](http://www.dbmetrics.nl).

Hierbij dank ik de volgende personen van harte voor hun inspirerende bijdrage aan dit boek en de fijne samenwerking!

- |                                   |  |
|-----------------------------------|--|
| • D. (Dennis) Boersen             | Argis IT Consultants                                       |
| • F. (Freek) de Cloe              | smartdocs.com  |
| • J.A.E. (Jane) ten Have          | -  |
| • Dr. L.J.G.T. (Louis) van Hemmen | BitAll B.V.  |
| • J.W. (Jan-Willem) Hordijk       | Cloud Advisor - Nordcloud, an IBM company                  |
| • W. (Willem) Kok                 | Argis IT Consultants                                       |
| • N (Niels) Talens                | <a href="http://www.nielstalens.nl">www.nielstalens.nl</a> |
| • D. (Dennis) Wit                 | ING  |

Ik wens u veel plezier toe bij het lezen van dit boek en vooral veel succes bij het toepassen van Continuous Design binnen uw eigen organisatie.

Mocht u vragen of opmerkingen hebben, aarzel dan vooral niet om met mij contact op te nemen. Er is veel tijd besteed om dit boek zo compleet en consistent mogelijk te maken. Mocht u toch tekortkomingen aantreffen, dan zou ik het op prijs stellen als u mij daarvan in kennis stelt, dan kunnen deze zaken in de volgende editie verwerkt worden.

# 1 Introductie

## Leeswijzer:

Dit hoofdstuk beschrijft het doel van dit boek (1.1), de doelgroep (1.2), de achtergrond van dit boek (1.3), de structuur (1.4), de bijlagen (1.5) en ten slotte enkele tips voor het hanteren van dit boek (1.6).

## 1.1 Doel

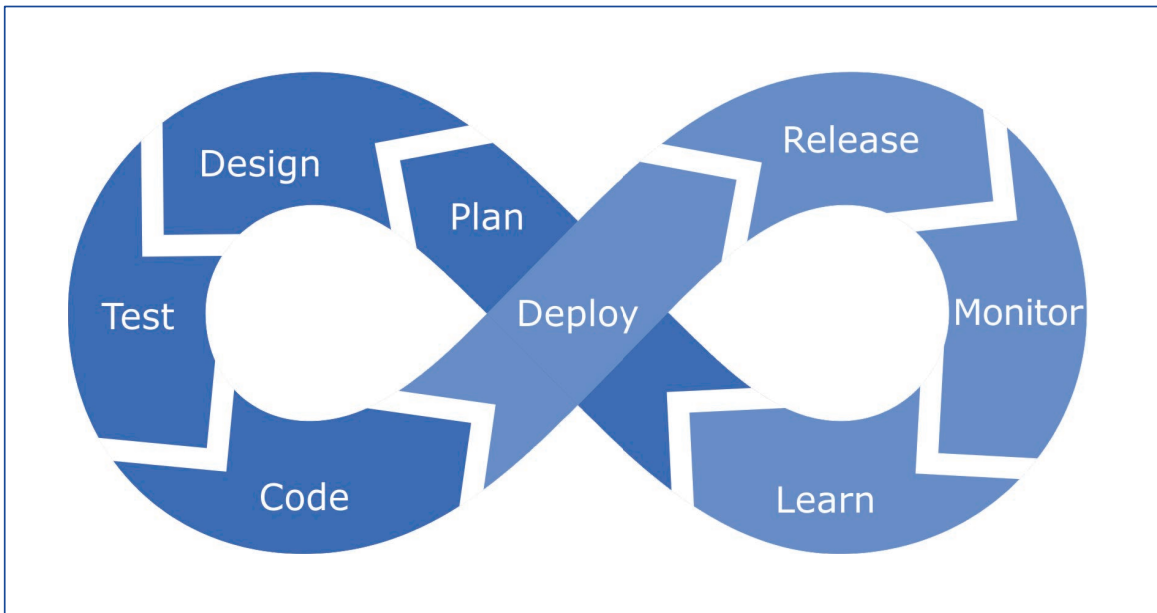
De doelstelling van dit boek is het geven van basiskennis ten aanzien van Continuous Design en tips en trucs voor het toepassen van dit aspectgebied van Continuous Everything.

## 1.2 Doelgroep

De doelgroep van dit boek zijn alle betrokken functies bij de DevOps teams. Dit omvat zowel de architecten, Dev engineers, Ops engineers, Product owners, Scrum masters, Agile Coaches en vertegenwoordigers van de gebruikersorganisatie. Dit boek is uiteraard ook zeer geschikt voor lijnmanagers, proceseigenaren, procesmanagers et cetera die betrokken zijn bij de totstandkoming van de informatievoorziening middels een DevOps werkwijze. Tenslotte is er een doelgroep die niet ontwikkelt of beheert maar die vaststelt of de informatievoorziening voldoet aan de vereiste criteria. Deze doelgroep omvat kwaliteitsmedewerkers en auditors. Zij kunnen dit boek gebruiken voor het identificeren van risico's die genomen of beheerst moeten worden.

## 1.3 Achtergrond

Dit boek bevat verschillende technieken om op een continue wijze invulling te geven aan een design dat meegroeit met het te ontwikkelen informatiesysteem. Continuous Design is een onderdeel van de DevOps Lemniscaat zoals weergegeven in [Figuur 1-1](#).



Figuur 1-1, DevOps Lemniscaat.

De DevOps lemniscaat geeft een overzicht van de te doorlopen fasen om software continue voort te brengen. De DevOps Lemniscaat is daardoor een goede basis voor het definiëren van het concept van Continuous Design.

Continuous Design is één van de aspectgebieden van het Continuous Everything concept. Het CE-concept beschrijft alle fasen van de DevOps Lemniscaat in de vorm van continue uit te voeren activiteiten. In [Tabel 1-1](#) is de relatie weergegeven tussen de stappen van de DevOps lemniscaat en de Continuous Everything aspectgebieden.

Development		Operations	
1	Continuous Planning (Plan)	6	Continuous Deployment (Release)
2	Continuous Design (Design)	7	Continuous Monitoring (Monitor)
3	Continuous Testing (Test)	8	Continuous Learning (Learn)
4	Continuous Integration (Code)	9	Continuous Auditing (-)
5	Continuous Deployment (Deploy)	10	Continuous Assessment (-)

Tabel 1-1, Continuous Everything aspecten.

Continuous Auditing (9) en Continuous Assessment (10) zijn niet in de DevOps Lemniscaat weergegeven, net als andere Continuous aspectgebieden zoals Continuous Documentation om wille van de eenvoud van de DevOps Lemniscaat.

Het woord 'Continuous' duidt op het incrementeel en iteratief ontwikkelen van software waardoor er een 'stroom' van code ontstaat die continue naar productie wordt overgezet door de CI/CD secure pipeline. Deze stroom is niets anders dan een value stream die geoptimaliseerd moet worden.

Continuous Design is een Agile wijze van vormgeven van een informatiesysteem door op het juiste tijdstip het gedrag, de functionaliteit en de kwaliteit van het informatiesysteem te definiëren. Alle aspecten van de DevOps Lemniscaat hebben direct of indirect een relatie met Continuous Design omdat deze holistisch wordt vormgegeven. Dit wil zeggen dat het Continuous Design betrekking heeft op zowel het informatiesysteem (Technology) al het voortbrengingsproces (Process) als de kennis en de kunde (People). Daarmee geeft Continuous Design dus een design op PPT nieuw.

Het bestaansrecht van een design (ontwerp) voor een informatiesysteem is de laatste jaren bij veel organisaties in twijfel getrokken. De klassieke rechtvaardiging van het bundelen van informatie over een informatiesysteem en het betrekken van alle stakeholders wordt gezien als achterhaald door de Agile manier van werken en het idee van 'the three amigo development strategy'. Deze strategie houdt in dat vanuit drie vakdisciplines: business, development en testen van tevoren gekeken wordt naar een increment dat moet worden gebouwd. Daarmee worden de 'hoe' en de 'wat' vraag beter uitgewerkt en kan er consensus bereikt worden over de Definition of Done (DoD) van het increment. Hierbij wordt echter voorbijgegaan aan de andere klassieke rechtvaardiging van een ontwerp en dat is dat een ontwerp ook bedoeld is voor kennisoverdracht, ondersteuning van service management en het nakomen van wet- en regelgeving.

Vanuit het gezichtspunt van service management, zijn de ontwikkelingen die binnen de DevOps wereld plaatsvinden, daarom ook erg belangrijk om te volgen. Waar er dit moment aan de ene kant nog steeds organisaties zijn die werken met watervalprojecten die een grote design inspanning vergen, zijn er ook al organisaties die ervaren dat het alleen werken met user stories niet zaligmakend is en dat enige vorm van design wel degelijk noodzakelijk is. En zo komt de wereld van systeemontwikkeling weer een beetje in balans.

De vraag is natuurlijk wel of er eenzelfde opzet van werken moet gelden voor alle typen van informatiesystemen. Met de komst van het BI-model van Gartner is het duidelijk geworden dat er een onderscheid gemaakt moet worden tussen de System of Records (SoR) en de System of Engagement (SoE) informatiesystemen. Naast de SoE wordt tegenwoordig ook wel gesproken over de System of Intelligence (SoI). In [Figuur 1-2](#) is een overzicht gegeven van de samenhang van de drie typen van informatiesystemen (SoR, SoE en SoI).

### System of Records

De SoR zijn informatiesystemen van de backoffice die de financiën, logistiek, voorraad en Human Resource Management (HRM) taken vervullen. Deze systemen staan onder toezicht van de overheid en moeten aan vele wet- en regelgevingseisen voldoen zoals die van de belastingdienst, De Nederlandse Bank (DNB) en de Autoriteit Financiële Markten (AFM). Maar ook de Sarbanes Oxley (SoX)-wetgeving en de Algemene Verordening Gegevensbescherming (AVG) zijn bronnen van eisen die een design van een informatiesysteem rechtvaardigen. De financiële informatiesystemen moeten daarenboven voldoen aan de General IT Controls (GITC) van de accountants om een handtekening te krijgen onder de jaarrekening.

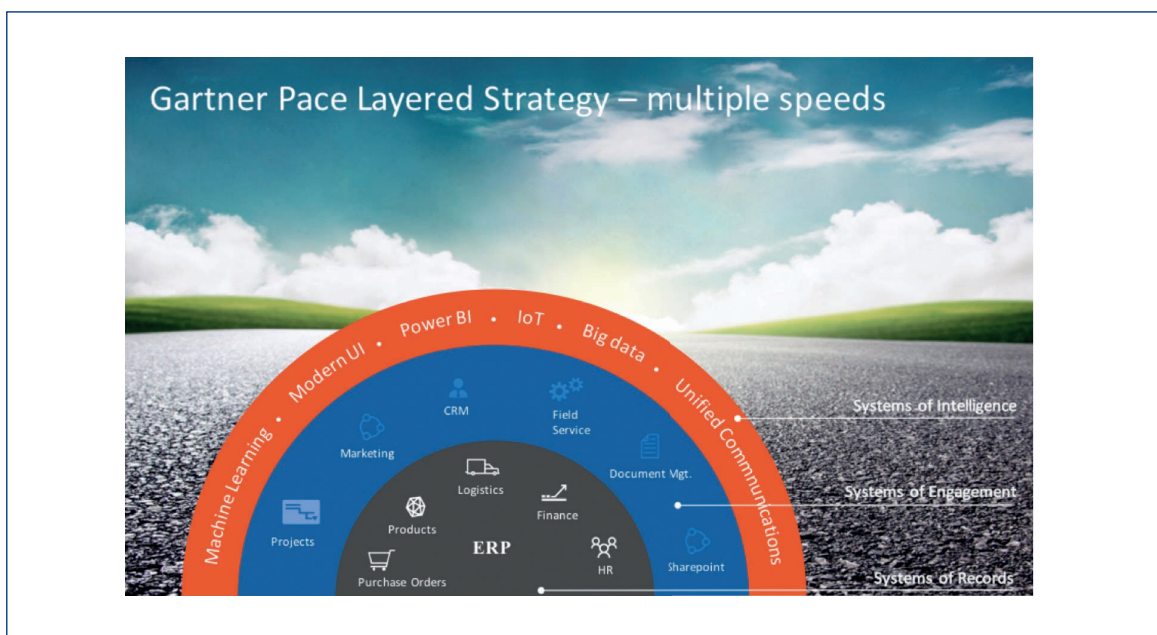
Dit betekent onder andere dat er designs vereist worden die aangeven hoe de financiële gegevens worden voortgebracht en wat de interfaces zijn van de verschillende betrokken informatiesystemen. Het zijn in het algemeen informatiesystemen die onderdeel zijn van een keten van informatiesystemen. Deze systemen vereisen een van tevoren goed overwogen aanpak en dus een design.

### System of Engagement

De SoE informatiesystemen zijn gericht op verkoopkanalen naar consumenten waarbij vooral gedacht moet worden aan webshops en apps voor de smartphones. Deze applicaties zijn eenvoudig te voorzien van een nieuwe release, versie en patch. Deze informatiesystemen zijn meestal geen integraal onderdeel van een keten maar veel eerder de eindpunten van een keten. Dit zijn vaak ook de voorbeelden uit de publicaties over Agile en Development & Operations (DevOps). Voor deze informatiesystemen is het duidelijk dat een van tevoren doordacht design (upfront design) minder hard nodig is en vaak volstaan kan worden met een groeiend design (emerging design). Wel blijkt dat voor deze SoE informatiesystemen het handig is, om meer dan alleen een verzameling user stories te hebben. Te meer omdat user stories vaak op de sprint backlog worden gezet en gearchiveerd worden na de realisatie van de sprint. Tevens vormen de losse user stories geen toegankelijke beschrijving van een informatiesysteem. Er is dus nog steeds behoefte aan een design dat overzicht en inzicht geeft van de functionaliteit, kwaliteit en de werking van het informatiesysteem.

### System of Intelligence

Daarnaast zijn er nog de Business Intelligence (BI) oplossingen. Dit zijn de rapportages, data-analyse tools en dergelijke. Voor deze toepassingen geldt hetzelfde als de SoE informatiesystemen. Ze zijn de weergave van informatie uit de SoR en zijn makkelijker aan te passen.



Figuur 1-2, SoR, SoE en SoI (source HSO the result company).

### Noodzaak Continuous Design

Voor alle drie de typen (SoR, SoE en SoI) informatiesystemen is dus een behoefte aan een Continuous Design. Er is meer dan alleen een set van user stories nodig om overzicht en inzicht te krijgen en te houden van het informatiesysteem. Anders kunnen de risico's en de impact van aanpassing en uitbreiding van het informatiesysteem niet tijdig en correct worden onderkend.

Evenwel moet voorkomen worden dat het design de Agility van het voortbrengingsproces te niet doet. Dit betekent dat het design incrementeel en iteratief moet worden vormgegeven (Continuous Design). De mate waarin het design van tevoren gedefinieerd moet zijn, verschuift van veel bij SoR naar minder bij SoE en nagenoeg niets bij SoI. Maar ook bij een SoR kan het design worden opgedeeld in lagen die meer upfront (van tevoren) en emerging (in sprints) gedefinieerd worden. Om dit uiteen te zetten wordt in dit boek eerst de Continuous Design Pyramid besproken dat een Continuous Design opdeelt in lagen (views) en daarna wordt elke view in een apart hoofdstuk uitgelegd.

## 1.4 Structuur

Dit boek omvat zes views om aan Continuous Design vorm te geven. Voordat deze worden besproken wordt eerst invulling gegeven aan de definities, de verankering en architectuur van Continuous Design. Daarna volgt per hoofdstuk de bespreking van de zes views.

### 1.4.1 Hoofdstuk 2: Basisconcepten en basisbegrippen

Dit hoofdstuk bespreekt de basisconcepten en de basisbegrippen.

### 1.4.2 Hoofdstuk 3: Continuous Design definitie

Het is belangrijk om een gemeenschappelijke definitie te hebben van Continuous Design. Daarom wordt er in dit hoofdstuk een definitie gegeven van dit concept en worden de problemen en mogelijke oorzaken van het niet goed ontwerpen van een informatiesysteem besproken.

### 1.4.3 Hoofdstuk 4: Continuous Design verankering

Dit hoofdstuk bespreekt hoe Continuous Design verankerd kan worden middels het veranderparadigma. Daarbij worden de volgende vragen beantwoord.

- Wat is de visie op Continuous Design (Beeldvorming)?
- Waar liggen de verantwoordelijkheden en bevoegdheden (Machtsverhouding)?
- Hoe kan Continuous Design worden toegepast (Organisatievormgeving)?
- Welke profielen van mensen en welke middelen zijn nodig (Resources)?

### 1.4.4 Hoofdstuk 5: Continuous Design Architectuur

Dit hoofdstuk beschrijft de architectuurprincipes en – modellen voor Continuous Design. De architectuurmodellen betreffen het 'Continuous Design Pyramid' model en het 'Continuous Design Planning' model.

### 1.4.5 Hoofdstuk 6: Continuous Design Ontwerp

Het Continuous Design ontwerp definieert de Continuous Design Value Stream en Use Case Diagram.

### 1.4.6 Hoofdstuk 7: Business View

De business view omvat de beschrijving van de bedrijfsprocessen aan de hand van een 'System Context Diagram' en een 'Value Stream Canvas model'.

### 1.4.7 Hoofdstuk 8: Solution View

De solution view omvat de beschrijving van het informatiesysteem op hoofdlijnen en tevens de wijze waarop het informatiesysteem gerelateerd is aan de bedrijfsprocessen en de stakeholders. Hiertoe worden drie modellen gebruikt te weten het 'Use Case Diagram', de 'System Building Blocks' en de 'Value Stream Mapping'.

### 1.4.8 Hoofdstuk 9: Design View

De design view is gebaseerd op het gebruik van Use Cases.

### 1.4.9 Hoofdstuk 10: Requirement View

De requirement view gebruikt de Behavior Driven Development (BDD) aanpak om het gedrag van het informatiesysteem te beschrijven.

### 1.4.10 Hoofdstuk 11: Test View

De test view is een onderdeel van het Continuous Design omdat op dit niveau de technische specificaties worden vastgelegd op unit testcase niveau, gebruikmakend van Test Driven Development (TDD).

### 1.4.11 Hoofdstuk 12: Code View

De code view is ook een onderdeel van het Continuous Design omdat op dit niveau de beslissingen worden genomen op welke wijze de technische requirements (unit testcases) geprogrammeerd dienen te worden. Op basis van de sourcecode en tevens alle andere deliverables van de deployment pipeline zoals infrastructure as code, testcases, deployment scripts, logfiles en dergelijk kan een applicatie design worden gegenereerd met behulp van gespecialiseerde tools. Omdat hierdoor de documentatie continue wordt gecreëerd en onderhouden spreekt men ook wel van Continuous Documentation.

#### 1.4.12 Hoofdstuk 13: Continuous Design bij Assuritas

Dit hoofdstuk geeft een voorbeeld van de toepassing van Continuous Design.

#### 1.4.13 Hoofdstuk 14: Continuous Design assessment

De volwassenheid van Continuous Design is in dit hoofdstuk meetbaar gemaakt aan de hand van een Continuous Design assessment.

## 1.5 Bijlagen

De bijlagen bevatten belangrijke informatie die helpt bij het beter begrijpen van Continuous Design.

Bijlagen	Onderwerp	Toelichting
A	Literatuurlijst	In dit boek wordt verwezen naar geraadpleegde literatuur in de vorm van: [Auteur Jaar]. In de bijlage zijn de volledige naam van de auteur, de titel en het ISBN-nummer weergegeven.
B	Begrippenlijst	Alleen de belangrijkste concepten worden in deze bijlage uitgelegd.
C	Afkortingen	Binnen de wereld van DevOps worden veel afkortingen gebruikt. Voor de leesbaarheid van dit boek zijn veel gebruikte termen afgekort. De eerste keer dat een afkorting wordt gebruikt is deze voluit geschreven.
D	Agile Tools	Tools zijn onlosmakelijk verbonden met DevOps. Deze bijlage geeft een overzicht van de in de praktijk veel gebruikte tools.
E	Websites	Een aantal relevante websites zijn in deze bijlage opgenomen. In dit boek wordt verwezen naar deze websites door de referentie: [http Name].
F	Index	De index omvat de verwijzing van termen die in dit boek zijn gehanteerd.

Tabel 1-2, Bijlagen.

## 1.6 Leeswijzer

In dit boek zijn het aantal afkortingen beperkt gehouden. Termen die echter steeds terugkomen zijn wel als afkorting weergegeven om de leesbaarheid te vergroten. Bijlage C geeft deze afkortingen weer.



# Bijlagen

## Bijlage A, Literatuurlijst

In [Tabel A-1](#) is een overzicht gegeven van boeken die direct of indirect zijn gerelateerd aan DevOps.

Referenties	Publicaties
Best 2011a	B. de Best, "SLA best practices", Leonon Media 2011, ISBN13: 978 90 71501 456.
Best 2011b	B. de Best, "ICT Performance-Indicatoren", Leonon Media 2011, ISBN13: 978 90 71501 470.
Best 2012	B. de Best, "Quality Control & Assurance", Leonon Media 2012, ISBN13: 978 90 71501 531.
Best 2014a	B. de Best, "Acceptatiecriteria", Leonon Media, 2014, ISBN 13: 978 90 71501 784.
Best 2014b	B. de Best, "Agile Service Management met Scrum", Leonon Media, 2014, ISBN13: 978 90 71501 807.
Best 2014c	B. de Best, "Cloud SLA, Leonon Media, 2014 ISBN13: 978 90 71501 739.
Best 2015a	B. de Best, "Agile Service Management met Scrum in de Praktijk", Leonon Media, 2015, ISBN13: 978 90 71501 845.
Best 2017a	B. de Best, "Beheren onder Architectuur", Dutch language, Leonon Media, 2017, ISBN13: 978 90 71501 913.
Best 2017b	B. de Best, "DevOps best practices", English language, Leonon Media, 2017, ISBN13: 978 94 92618 078.
Best 2017c	B. de Best, "SLA Templates", English language, Leonon Media, 2017, ISBN13: 978 94 92618 320.
Best 2019b	B. de Best, "DevOps Architectuur", Dutch language, Leonon Media, 2019, ISBN13: 978 94 92618 061.
Best 2021b	B. de Best, "Basiskennis IT", Dutch language, Leonon Media, 2021, ISBN13: 978 94 92618 573.
Best 2022 CA	B. de Best, "Continuous Auditing", Dutch language, Leonon Media, 2022, ISBN13: 978 94 92618 542.
Best 2022 CD	B. de Best, "Continuous Deployment", Dutch language, Leonon Media, 2022, ISBN13: 978 94 92618 511.
Best 2022 CI	B. de Best, "Continuous Integration", Dutch language, Leonon Media, 2022, ISBN13: 978 94 92618 467.
Best 2022 CL	B. de Best, "Continuous Learning", Dutch language, Leonon Media, 2022, ISBN13: 978 94 92618 528.
Best 2022 CM	B. de Best, "Continuous Monitoring", Dutch language, Leonon Media, 2022, ISBN13: 978 94 92618 498.
Best 2022 CN	B. de Best, "Continuous Design", Dutch language, Leonon Media, 2022, ISBN13: 978 94 92618 481.
Best 2022 CP	B. de Best, "Continuous Planning", Dutch language, Leonon Media, 2022, ISBN13: 978 94 92618 504.
Best 2022 CS	B. de Best, "Continuous Assessment", Dutch language, Leonon Media, 2022, ISBN13: 978 94 92618 474.
Best 2022 CT	B. de Best, "Continuous Testing", Dutch language, Leonon Media, 2022, ISBN13: 978 94 92618 450.

Referenties	Publicaties
Best 2022 CY	B. de Best, "Continuous Security", Dutch language, Leonon Media, 2022, ISBN13: 978 94 91480 171.
Best 2022a	B. de Best, "Continuous Development", Dutch language, Leonon Media, 2022, ISBN13: 978 94 92618 641.
Best 2022b	B. de Best, "Continuous Operations", Dutch language, Leonon Media, 2022, ISBN13: 978 94 92618 658.
Best 2022c	B. de Best, "Continuous Control", Dutch language, Leonon Media, 2022, ISBN13: 978 94 91480 195.
Best 2022d	B. de Best, "Continuous Everything", Dutch language, Leonon Media, 2022, ISBN13: 978 94 92618 597.
Bloom 1956	Benjamin S. Bloom, "Taxonomy of Educational Objectives (1956)", Allyn and Bacon, Boston, MA. Copyright (c) 1984 by Pearson Education.
Boehm	Boehm B. Software Engineering Economics, Prentice Hall, 1981
Caluwé 2011	L. de Caluwé en H. Vermaak, "Leren Veranderen", Kluwer, 2011, tweede druk, ISBN13: 978 90 13016 543.
Davis 2016	Jennifer Davis, Katherine Daniels, "Effective DevOps Building a Culture of Collaboration, Affinity, and Tooling at Scale", O'Reilly Media; 1 edition, ISBN-13: 978 14 91926 307, 2016.
Deming 2000	W. Edwards Deming, "Out of the Crisis. MIT Center for Advanced Engineering Study", 2000, ISBN13: 978 02 62541 152.
Downey 2015	Allen. B. Downey, "Think Python", O'Reilly Media, Inc, Usa; Druk 2, ISBN-13: 978 14 91939 369, 2015.
Galbraith 1992	Galbraith, J.R. "Het ontwerpen van complexe organisaties", 1992, Alphen aan de Rijn: Samson Bedrijfsinformatie.
Humble 2010	Jez Humble, David Farley "Continuous Delivery Reliable Software Releases through Build, Test, and Deployment Automation", Addison-Wesley Professional; 1 edition, ISBN-13: 978 03 21601 919, 2010.
Kim 2014	Gene Kim, Kevin Behr, George Spafford "The Phoenix Project", IT Revolution Press, ISBN-13: 978 09 88262 508, 2014.
Kim 2016	Gene Kim, Jez Humble "The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, Patrick Debois, John Willis", IT Revolution Press, ISBN-13: 978 19 42788 003, 2016.
Kotter 2012	John P. Kotter, "Leading Change", Engels 1e druk, november 2012, ISBN13: 978 14 22186 435.
Kaplan 2004	R. S. Kaplan en D. P. Norton, "Op kop met de Balanced Scorecard", 2004, Harvard Business School Press, ISBN13: 978 90 25423 032.
Layton 2017	Mark C. Layton Rachele Maurer, "Agile Project Management for Dummies", tweede druk, John Wiley & Sons Inc, 2017, ISBN13: 978 11 19405 696.
Looijen 2011	M. Looijen, L. van Hemmen, "Beheer van Informatiesystemen", zevende druk, Academic Service, 2011, ISBN13: 978 90 12582 377.
MAES	R. Maes, "Visie op informatiemanagement", www.rikmaes.nl.
McCabe	McCabe T. "A Complexity Measure" in: IEEE Transactions on Software Engineering 1976, vol. 2, nr. 4.
Michael Porter 1998	M.E. Porter 'Competitive Advantage: Creating and Sustaining Superior Performance, Simon & Schuster, 1998, ISBN13: 978 06 84841 465.

Referenties	Publicaties
Oirsouw 2001	R.R. van Oirsouw, J. Spaanderman, C. van Arendonk, "Informatiseringseconomie", ISBN 90 395 1393 7, 2001.
Scrum	Ken Schwaber and Jeff Sutherland, "The Scrum Guide™", 2017, <a href="http://www.scrumguides.org">www.scrumguides.org</a> .
Schwaber 2015	K. Schwaber, "Agile Project Management with Scrum", Microsoft Press, ISBN13: 978 07 35619 937.
Toda 2016	(Luke) Toda, President Strategic Staff Services Corporation and Director of TPS Certificate Institution Nobuyuki Mitsui, CTO of Strategic Staff Services Corporation, "Success with Enterprise DevOps Koichiro" "White Paper", 2016.

Tabel A-1, Literatuurlijst.

## Bijlage B, Begrippenlijst

In [Tabel B-1](#) is een begrippenlijst opgenomen. Deze is in het Engels opgesteld omdat zeer veel termen uit de Engelse taal afkomstig zijn en de uitleg makkelijker leest als de hele uitleg in het Engels is opgenomen.

Begrip	Betekenis
5S	Japan's principle of order and cleanliness. These Japanese terms with their Dutch equivalent are: Seiri (整理): Sort Seiton (整頓): Arrange Seisō (清掃): Cleaning Seiketsu (清潔): Standardise Shitsuke (躰): Hold or Systematise <a href="#">[Wiki]</a>
<a href="#">A/B testing</a>	A/B testing means that two versions of an application or webpage are taken into production to see which performs better. Canary releasing can be used, but there are also other ways to perform A/B testing.
<a href="#">Acceptance test</a>	For DevOps engineers the acceptance testcases gives the answer "How do I know when I am done?". For the users the acceptance testcases gives the answer "Did I get what I wanted?". Examples of acceptance testcases are Functional Acceptance Testcases ( <a href="#">FAT</a> ), User Acceptance Testcases ( <a href="#">UAT</a> ) and Production Acceptance Testcases ( <a href="#">PAT</a> ). The FAT and UAT should be expressed in the language of the business.
<a href="#">Affinity</a>	DevOps is about <a href="#">collaboration</a> and affinity. Where collaboration is focused on the relationship between individuals in a DevOps team, affinity goes one step further. This DevOps pillar is about shared organisational goals, empathy and learning between different groups of people by sharing stories and learn from each other.
<a href="#">Agile Infrastructure</a>	Within DevOps both Development and Operations work in an Agile way. This requires an Agile Infrastructure that can be changed with the same pace as the application is changed through the deployment pipeline. A good example of an Agile Infrastructure is the use of Infrastructure as Code.
<a href="#">Alternate path</a>	See <a href="#">happy path</a> .
<a href="#">Andon cord</a>	In the Toyota manufacturing plant, above every work centre a cord is installed. Every worker and manager are trained to pull when something goes wrong; for example, when a part is defective, when a required part is not available, or even when work takes longer than planned.  When the Andon cord is pulled, the team leader is alerted and immediately works to resolve the problem. If the problem cannot be resolved within a specified time (e.g., fifty-five seconds), the production line is stopped so that the entire organisation can be mobilised to assist with problem resolution until a successful countermeasure has been developed <a href="#">[Kim 2016]</a> .
<a href="#">Anomaly detection techniques</a>	Not all data that needs to be monitored has a Gaussian (normal) distribution. The anomaly detection techniques make it possible to find noteworthy variances using a variety of methods for data that has no Gaussian distribution. These techniques are either used in monitoring tools or require people with statistical skills.

Begrip	Betekenis
Anti-pattern	An anti-pattern is an example of the wrong interpretation of a <u>pattern</u> . The anti-pattern is often used to explain the value of the <u>pattern</u> .
Antifragility	This is the process of applying stress to increase resilience. This term is introduced by author and risk analyst Nassim Nicholas Taleb.
Artefact	An artefact is a product that is manufactured. Within DevOps the output of the commit phase are binaries, reports and meta data. These products are also referred to as artefacts.
Artefact repository	The central storage of artefacts is called the artefact repository. The artefact repository is used to managed artefacts and their dependencies.
Automated tests	Testcases should be automated as much as possible to reduce waste and to increase velocity and quality of the products that are to be delivered.
Bad apple theory	People that believe in the 'Bad Apple Theory' think that a system is basically safe if it were not for those few unreliable people in it. By removing these people, the system will be safe. This results in the anti DevOps pattern of 'name, blame, shame'.
Bad paths	A 'bad path' is a situation where the application does not follow the 'happy path' or 'the alternate' path. In other words, something goes wrong. This exception must be handled and should be monitorable.
Behavior Driven Development (BDD)	The development of software requires that the users are asked to define the (non) functional requirements. Behavior driven development is based on this concept. The difference however is that the acceptance criteria of these requirements should be written in the customer's expectation of the behavior of the application. This can be accomplished by formulating the acceptance criteria in the <u>Given - When - Then</u> format.
Binary	A compiler is used to transform source code to object code. The object code is also known as a binary. The source code is readable for human being, the object code however is only readable for computers since they have been written in hexadecimals.
Blameless post-mortem	Blameless post-mortem is a term coined by John Allspaw. It helps to examine "mistakes in a way that focuses on the situational aspects of a failure's mechanism and the decision-making process of individuals proximate to the failure." [Kim 2016].
Blamelessness	This approach is about learning rather than punishing. Within DevOps this is one of the basic ideas of learning from mistakes. The energy of the DevOps team is spending on learning from the mistake, rather than on finding the one to blame.
Blue-Green deployment pattern	Blue and green refer to two identical production systems. One is used for the final acceptance of a new release. If this acceptance is successful, then this environment becomes the new production environment. In case of a failure of the production system, the other system can be used instead. This mitigates the risk of downtime since the switchover is likely to be less than a second.
Broken build	A build that fails due to an error in the application source code.
Brown field	There are two scenarios' for applying DevOps best practices: green field and brown field. In case of a green field scenario the whole DevOps organisation has to be established from scratch. The opposite scenario is where there is already a DevOps organisation, but improvements are needed.

Begrip	Betekenis
	The colour green refers to the situation that a factory is built on a clean grass field. The colour brown refers to the situation that a factory is to be built on a place where there has already been a factory that poisoned the ground. In order to build on a brown field, the poison needs to be removed.
Business value	Applying DevOps best practices results in increasing the business value. Research of Puppet Labs (State Of DevOps Report) proves that high-performing organisations using DevOps practices are outperforming their non-high performing peers in many following areas [Kim 2016].
Canary releasing pattern	Normally a release is offered to every user at once. Canary releasing is the approach in which a small set of users is receiving the new release. If this small scope release works fine than the release can be deployed to all users. The term canary refers to the old habit to have a canary in the coal mines to detect toxic gas.
Change categories	Changes can be categorised into standard changes, normal changes and urgent changes.
Change schedules	Changes can be scheduled in order to defined in which order they have to be applied.
Cloud configuration files	Cloud configuration files are used to initiate a cloud service before using it. In this way cloud service providers enable customers to configure the cloud environment for their needs.
Cluster immune system release pattern	The cluster immune system expands upon the <u>canary release pattern</u> by linking our production monitoring system with our release process and by automating the roll back of code when the user-facing performance of the production system deviates outside of a predefined expected range, such as when the conversion rates for new users drops below our historical norms of 15%–20% [Kim 2016].
Code branch	See <u>branching</u> .
Code review methods	Code review can be performed in several ways like “ <u>over the shoulder</u> ”, <u>pair programming</u> , <u>email pass-around</u> and <u>tool-assisted code review</u> .
Codified NFR	A list of Non-Functional Requirements (NFR) that are categorised in categories like availability, capacity, security, continuity et cetera.
Collaboration	One of the four pillars of DevOps is collaboration. Collaboration refers to the way the individuals of a DevOps team works together to achieve the common goal. There are many forms in which this collaboration comes to expression like: <ul style="list-style-type: none"> <li>• peer to peer programming;</li> <li>• demonstrating weekly progress;</li> <li>• documentation;</li> </ul> et cetera.
Commit code	Committing code is the action in which the DevOps engineer adds the changed source code to the repository, making these changes part of the head revision of the repository [Wiki].
Commit stage	This is the phase in the CI/CD secure pipeline where the source code is compiled to the object code. This includes the performance of the unit testcases.

Begrip	Betekenis
Compliance checking	The manual action of a security officer to make sure that the system is built in accordance with the agreed standards. This is the opposite of security engineering where the DevOps teams works together with the security officer in order to embed the agreed standards in the deliverables and enable continuous monitoring of the standard in the whole lifecycle of the product.
Compliance officer	The compliance officer is a DevOps role. The compliance officer is responsible for ensuring compliance with agreed standards throughout the whole life cycle of a product.
Configuration management	Configuration Management refers to the process by which all artefacts, and the relationships between them, are stored, retrieved, uniquely identified and modified.
Containers	A container is an isolated structure that is used by DevOps engineers to build their application independently from the underlying operating system or hardware. This is accomplished by interfaces in the container that are used by DevOps engineers. Instead of installing the application in an environment, the complete container is deployed. This saves a lot of dependencies and prevents configuration errors to occur.
Conway's law	The following statement of Melvin Conway is called the Conway's law: "organisations which design systems ... are constrained to produce designs which are copies of the communication structures of these organisations." [Wiki].
Cultural debt	There are three forms of debt. Cultural debt, <u>technical debt</u> and <u>information debt</u> . This form of debt refers to the decision to keep flaws in the organisation structure, hiring strategy, values et cetera. This debt costs interest and will result in less maturity growth of the DevOps teams. Cultural debt can be recognised by the exitance of extensive silos, workflow constraints, miscommunications, waste et cetera.
Culture, Automation Measurement, Sharing (CAMS)	CAMS is the abbreviation for Culture, Automation, Measurement and Sharing. <ul style="list-style-type: none"> <li>• Culture: Culture relates to the people and process aspects of DevOps. Without the right culture, automation attempts will be fruitless.</li> <li>• Automation: Release management, configuration management, and monitoring and control tools should enable automation.</li> <li>• Measurement: 'If you can't measure it, you can't manage it.' &amp; 'If you can't measure it, you can't improve it'.</li> <li>• Sharing: Culture of sharing ideas and problems is critical to help organisations to improve. Creates feedback loop.</li> </ul>
Cycle time (flow time)	Cycle time measures more the completion rate or the work capability of a system overall, and a shorter cycle time means that less time is being wasted when a request has been made but no progress or work is getting done.
Cycle time (lean)	The average time between two successive units leaving the work or manufacturing process.
Declarative programming	This is a <u>programming paradigm</u> that expresses the logic of a computation without describing its control flow. An example are the database query languages for example TSQL and PSQL.



Begrip	Betekenis
Defect tracking	Defect tracking is the process of tracking the logged defects in a product from beginning to closure and making new versions of the product that fix the defects [Wiki].
Development	Development is an activity that is performed by the DevOps role 'DevOps engineer'. A DevOps engineer is responsible for the complete lifecycle of a configuration item. Within DevOps there is no difference anymore between designer, builder or tester.
Development rituals	The Agile Scrum rituals of development are the sprint planning, daily stand-up, sprint execution, review and the retrospective.
Downward spiral	Gene Kim explains in his book [Kim 2016] that the downward spiral in Information Technology (IT) has three acts. <ul style="list-style-type: none"> <li>• The first act begins in IT Operations where technical debt results in jeopardising our most important organisational promises.</li> <li>• The second act starts with compensating the latest broken promise by promising a bigger, bolder feature or an even larger revenue target. As a result, Development is tasked with another urgent project which results in even more technical debt.</li> <li>• The third stage is where the deployments are getting slower and slower, and outages are increasing. The business value continuously decreases.</li> </ul>
E-mail pass-around	E-mail pass-around is a review technique where the source code management system emails code to reviewers automatically after the code is checked in [Kim 2016].
Error path	See <u>happy path</u> .
Fast feedback	Fast feedback refers to the second way of the three ways of Gene Kim. The second way is about having feedback on the functionality and quality of the product that is created or modified as soon as possible in order to maximise the business value.
Feature toggles	A feature toggle is a mechanism that makes it possible to enable or disable a part of the functionality of an application released in production. Feature toggles enables testing the effect of changes on users in production. Feature Toggles are also referred to as Feature Flags, Feature Bits or Feature Flippers.
Feedback	Feedback within the context of DevOps is the mechanism by which errors in the value stream are detected as soon as possible and is used to improve the product and if necessary to improve the value stream as well.
Feedforward	Feedforward within the context of DevOps is the mechanism by which experiences in the present value stream are used to improve the future value stream. Feed forward is the opposite of feedback since feedback is focused on the past and feed forward on the future.
Gaussian distribution	In probability theory, the normal (or Gaussian) distribution is a very common continuous probability distribution. Normal distributions are important in statistics and are often used in the natural and social sciences to represent real-valued random variables whose distributions are not known. A random variable with a Gaussian distribution is said to be normally distributed and is called a normal deviate [Wiki].

Begrip	Betekenis
Given-When-Then	The Given-When-Then format is used to define acceptance criteria in a way that the stakeholders understand how the functionality actually will work. GIVEN – the fact that... WHEN – I do this... THEN – this happens...
Green field	See brown field.
Hand-off Readiness Review (HRR)	The HRR term is introduced by Google. An HRR is set of safety checks for a critical stage of releasing new services. HRR is performed when a service is transitioned from a developer-managed state to an OPS-managed state (usually months after the LRR). HRR makes service transition easier and more predictable and helps create empathy between upstream and downstream work centers.
Happy path	An application supports a business process by receiving, editing, storing and providing information. The assumed steps in which the information processing is performed is called the happy path. The steps in alternate ways are called the alternate path. In that case, the same result will be achieved via another navigation path. The crawl of the application that causes an error is called an error path.
Holocracy	In this type of organisation all decisions are made through self-organising teams rather than through a traditional management hierarchy.
Horizontal splitting of features	A feature can be splitted into stories. Horizontal splitting refers to the result of a feature splitting in which more DevOps teams must work tightly together. They have to align their work continuously in order to deliver together the feature.
I-shaped, T-shaped, E-shaped	I-shaped, T-shaped, E-shaped are the categories to indicate the knowledge and special skills of a person. An I-shaped person is a pure specialist in one area. The T-shaped person has special skills in one field and broad general knowledge. The E-shaped person has special skills in more than one field and broad general knowledge.
Idempotent	Continuous delivery requires that a component can always to be brought fully automatically to the desired status regardless of the component's initial state and regardless of the number of times the component is configured. The characteristic of a component to always be able to get back into the desires is called idempotent.
Imperative programming	This is a <u>programming paradigm</u> that uses statements that change a program's state. Imperative programming focuses on how a program should operate and consists of commands for the computer to perform. Examples are COBOL, C, BASIC et cetera.  The term is often used in contrast to <u>declarative programming</u> , which focuses on what the program should accomplish without specifying how the program should achieve the result.
Independent, Negotiable, Valuable, Estimable, Small, and Testable (INVEST)	Independent, Negotiable, Valuable, Estimable, Small, and Testable. <ul style="list-style-type: none"> <li>• <b>Independent:</b> The product backlog item should be self-contained, in a way that there is no inherent dependency on another product backlog item.</li> <li>• <b>Negotiable:</b> Product backlog items, up until they are part of an iteration, can always be changed, rewritten or even discarded.</li> <li>• <b>Valuable:</b> Product backlog item must deliver value to the stakeholders.</li> </ul>

Begrip	Betekenis
	<ul style="list-style-type: none"> <li>• <b>Estimable:</b> The size of a product backlog item must always estimable.</li> <li>• <b>Small:</b> Product backlog items should not be so big as to become impossible to plan / task / prioritise with a certain level of certainty.</li> <li>• <b>Testable:</b> The product backlog item or its related description must provide the necessary information to make test development possible.</li> </ul>
Information radiators	An Information Radiator is a visual display that a team places in a highly visible location so that all team members can see the latest information at a glance.
Infosec	A team that is responsible for securing systems and data.
Infrastructure as Code (IaC)	Normally infrastructure components have to be configured in order to perform the requested functionality and quality for example a rule set for a firewall or the allowed IP addresses for a network. These configurations normally are stored in configuration files which enable the operators to manage the functionality and the quality of the infrastructure components. Infrastructure as code (IaC) makes it possible to programme these infrastructure component settings and deploy these settings through the CI/CD secure pipeline by the use of machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.
Infrastructure as Code (IaC)	Infrastructure as code (IaC) is a software-based approach to the ICT infrastructure, whereby the systems can be rolled out and adapted in a consistent manner through templates. If a change has to be made, it is implemented in the template which is then rolled out again.
Infrastructure management	Infrastructure management consists of the lifecycle management of all infrastructure products and services in order to support the correct working of the applications that run on top of the infrastructure.
Ji-Kotei-Kanketsu (JKK)	JKK which means 100% completion of an item. This quality way of working means: <ul style="list-style-type: none"> <li>• clear understanding of the goals;</li> <li>• understanding the right way to work;</li> <li>• ensure high quality of work;</li> <li>• getting the work right for 100% completion, never pass defects to the next process;</li> <li>• Definition of Done (DoD) is vital;</li> </ul> and then maintaining the required quality without inspections.
Just In Time (JIT)	JIT means building up a stream-lined supply chain with one-piece flow.
Kaizen	<p>Kaizen is Japanese for "improvement". Kaizen is used to improve production systems. The goals of kaizen are:</p> <ul style="list-style-type: none"> <li>• elimination of waste (<u>muda</u>'s);</li> <li>• <u>JIT</u>;</li> <li>• standardisation of production;</li> <li>• cycle of continuous improvements.</li> </ul> <p>Continuous improvement means circulate the Plan-Do-Check-Act (PDCA) cycle daily, weekly.</p> <p>This can be accomplished by finding the root cause of a failure by asking "Why" 5 times. The following steps can be followed:</p> <ul style="list-style-type: none"> <li>• defining problems with supporting data;</li> <li>• making sure everybody recognises the problems clearly;</li> </ul>

Begrip	Betekenis
	<ul style="list-style-type: none"> <li>• setting a hypothesis on the problems found;</li> <li>• defining countermeasure actions to verify the hypothesis;</li> <li>• defining countermeasure actions be in daily based activities;</li> <li>• measuring a weekly KPI so people can feel a sense of accomplishment.</li> </ul>
Kaizen Blitz (or Improvement Blitz)	A Kaizen Blitz is a rapid improvement workshop designed to produce results / approaches to discrete process issues within a few days. It is a way for teams to carry out structured, but creative problem solving and process improvement, in a workshop environment, over a short timescale.
Kaizen in advance	Kaizen in advance goes one step further than Kaizen. Not only the own activities are improved but also the activities that are performed upstream and that lead to problems downstream. In this way a feedback loop of problems is created which improves the system as a whole.
Kanban	<p>This is system to signal when something is needed. Kanban is a system for managing the logistics production chain. Kanban was developed by Taiichi Ohno, at Toyota, to find a system that made it possible to achieve a high level of production.</p> <p>Kanban is often used for application management. One of the characteristics of Kanban is that it is pull oriented which means that there is not stock of material to be used during the production. Kanban can be used to implement <u>JIT</u> in production systems.</p>
Kata	<p>A kata is any structured way of thinking and acting (pattern of behavior) that is practiced until the pattern becomes a second nature.</p> <p>Four steps can be recognised to accomplish this second nature:</p> <ul style="list-style-type: none"> <li>• direction (target);</li> <li>• current condition (IST situation);</li> <li>• target condition (SOLL situation);</li> <li>• PDCA (Deming wheel).</li> </ul> <p>From an architectural viewpoint the migration path might be added to Kata as well. The migration path shows the way to go in order to achieve the SOLL situation.</p>
Kibana dashboards	A Kibana dashboard displays a collection of saved visualisations.
Latent defects	Problems that are not visible yet. Latent defects can be made visible by injecting faults into the system.
Launch Readiness Review (LRR)	The LRR term is introduced by Google. An LRR is a set of safety checks for a critical stage of releasing new services. It is performed and signed off before a service is made publicly available and receive live production traffic. LRR is self-reported by the project teams. LRR is used in the development-managed state.
Launching guidance	To prevent the possibility of problematic, self-managed services going into production and creating organisational risk, launch requirements may be defined that must be met in order for services to interact with real customers and be exposed to real production traffic [Kim 2016].
Lead Time (LT)	Lead time is the time from when a request is made to when the final result is delivered, or the customer's point of view on how long something takes to complete.
Lean tools	<ul style="list-style-type: none"> <li>• A3 thinking (problem solving)</li> <li>• Continuous flow (eliminates waste)</li> </ul>

Begrip	Betekenis
	<ul style="list-style-type: none"> <li>• <a href="#">Kaizen</a></li> <li>• <a href="#">Kanban</a></li> <li>• KPI (Key Performance Indicator)</li> <li>• Plan Do Check Act (PDCA)</li> <li>• Root cause analysis</li> <li>• Specific, Measurable, Accountable, Realistic, Timely (SMART)</li> <li>• <a href="#">Value stream mapping</a> (depict the flow)</li> <li>• <a href="#">JJK</a> (No defects are passed to next process)</li> </ul>
Learning culture	<p>A learning culture is a collection of organisational conventions, values, practices and processes. These conventions encourage employees and organisations to develop knowledge and competence.</p> <p>An organisation with a learning culture encourages continuous learning and believes that systems influence each other. Since constant learning elevates an individual as a worker and as a person, it opens opportunities for the establishment to transform continuously for the better.</p>
Light weight ITSM	<p>This variant of Information Technology (IT) Service Management (<a href="#">ITSM</a>) is strictly focused on business continuity with a set of Minimum Required Information (MRIs). The MRI set for each organisation depends on their business.</p>
Logging levels	<p>Within monitoring systems there are several levels of logging recognised:</p> <ul style="list-style-type: none"> <li>• Debug level: Information at this level is about anything that happens in the program, most often used during debugging.</li> <li>• Info level: Information at this level consists of actions that are user-driven or system specific.</li> <li>• Warn level: Information at this level tells us of conditions that could potentially become an error.</li> <li>• Error level: Information at this level focuses on error conditions.</li> <li>• Fatal level: Information at this level tells us when we must terminate.</li> </ul>
Loosely coupled architecture	<p>Loosely coupled architectures enables that changes can be made safely and with more autonomy, increasing developer productivity.</p>
Micro service	<p>Microservices are a variant of the service-oriented architecture (SOA) architectural style that structures an application as a collection of loosely coupled services.</p> <p>In a microservices architecture, services should be fine-grained, and the protocols should be lightweight <a href="#">[Wiki]</a>.</p>
Micro service architecture	<p>This architecture consists of a collection of services where each service provides a small amount of functionality, and the total functionality of the system is derived from composing multiple versions of a service in production simultaneously and to roll back to a prior version relatively easily.</p>
Mini pipeline	<p>In rare cases more than one deployment pipeline is required in order to produce the entire application. This can be accomplished by the use of a pipeline per application component.</p> <p>All these components are then assembled in a central pipeline which puts the entire application through acceptance tests, non-functional tests, and then deploys the entire application to testing, staging, and production environments.</p>

Begrip	Betekenis
Monitoring Framework	A framework of components that together form a monitor facility that is capable to monitor business logic, applications, and operating systems. Events, logs and measures are routed by the event router to destinations [Kim 2016].
Monolithic	A monolithic architecture is the traditional programming model, which means that elements of a software program are interwoven and interdependent. That model contrasts with more recent modular approaches such as a micro service architecture (MSA).
MTTR	Mean Time To Repair (MTTR) is a basic measure of the maintainability of repairable items. It represents the average time required to repair a failed component or device.
Muda	This is a Japanese word for waste. It is used in relationship to production systems.
Non-Functional Requirement (NFR)	NFR are requirements that define the quality of a product like maintainability, manageability, scalability, reliability, testability, deploy ability and security. NFR are also referred to as operational requirements.
Non-Functional Requirement (NFR) testing	NFR testing is the testing aspect that focusses on the quality of the product.
Obeya	Obeya is a war room which serves two purposes: <ul style="list-style-type: none"> <li>• information management;</li> <li>• and on-the-spot decision making.</li> </ul>
One piece flow	The Lean approach means that the DevOps team only works at one item at a time as a team with a fast pace and smooth flow. This is also used in the first way of the three ways of Gene Kim.
Operations	Operations is the team often responsible for maintaining the production environment and helping to ensure that required service levels are met [Kim 2016].
Operations stories	The work that has to be done by Ops can be written in stories. In that way that can be prioritised and managed.
OPS liaison	An OPS liaison is an operation employee who is assigned to a development team in order to facilitate the development team for their infrastructural demands.
Organisation archetypes	There are three organisation archetypes: functional, matrix, and market. They are defined by Dr. Roberto Fernandez as follows: <ul style="list-style-type: none"> <li>• Functional: Functional-oriented organisations optimise for expertise, division of labour, or reducing cost.</li> <li>• Matrix: Matrix-oriented organisations attempt to combine functional and market orientation.</li> <li>• Market: Market-oriented organisations optimise for responding quickly to customer needs.</li> </ul>
Organisational typology model	This a model of Dr. Ron Westrum in which he defined three types of culture: 'pathological', 'bureaucratic', 'generative'. These organisation types can be recognised by the following characteristics: <ul style="list-style-type: none"> <li>• Pathological organisations are characterised by large amounts of fear and threat.</li> <li>• Bureaucratic organisations are characterised by rules and processes.</li> <li>• Generative organisations are characterised by actively seeking and sharing information to better enable the organisation to achieve its mission.</li> </ul>

Begrip	Betekenis
	Dr. Westrum observed that in healthcare organisations, the presence of “generative” cultures was one of the top predictors of patient safety.
Over-the-shoulder	This is a review technique where the author walks through his code while another developer gives feedback.
Packages	A set of individual files or resources which are packed together as a software collection that provides certain functionality as part of a larger system.
Pair-programming	This is review technique where two developers work together using one computer. While one developer writes the code the other reviews it. After one hour they exchange their role.
Peer review	This is a review technique where developers review each other’s code.
Post-mortems	After a major incident a post-mortem meeting can be organised in order to find out what the root-cause is of the incident and how to prevent it in the future.
Product owner	The Product Owner is a DevOps role. The Product Owner is the internal voice of the business. The Product Owner is the owner of the product backlog and determines the priority of the product backlog items in order to define the next set of functionalities in the service.
Programming paradigm	A style of building the structure and elements of computer programs.
Pull request process	This is a form of peer review that span Dev and Ops. It is the mechanism that lets engineers tell others about changes they have pushed to a repository.
Quality Assurance (QA)	Quality Assurance (QA) is the team responsible for ensuring that feedback loops exist to ensure the service functions as desired [Kim 2016].
Reduce batch size	The size of a batch has an influence on the flow. Small batch sizes results in a smooth and fast flow. Large batch sizes results in high Work In Progress (WIP) and increases the level of variability in flow.
Reduce number of handoffs	In terms of a software process a handoff means that the work that is performed in order to produce software is stopped and handed over to another team. Each time the work passes from one team to another team, this requires all sorts of communication using different tools and filling up queues of work. To less handoffs the better.
Release managers	This a DevOps role. The release manager is responsible for managing and coordinating the production deployment and release processes.
Release patterns	There are two patterns of releases to be recognised [Kim 2016]: <ul style="list-style-type: none"> <li>• Environment-based release patterns: In this pattern there are two or more environments that receive deployments, but only one environment is receiving live customer traffic.</li> <li>• Application-based release patterns: In this pattern the application is modified in order to make selectively releases possible and to expose specific application functionality by small configuration changes.</li> </ul>
Sad path	A specific type of a ‘ <u>bad path</u> ’ is called a ‘sad path’. This is the case if the ‘bad path’ results in a security-related error condition.
Safety checks	Safety checks are performed during a release of a product. They are typical part of an <u>HRR</u> of an <u>LRR</u> .

Begrip	Betekenis
SBAR	This technique offers guidelines for making sure concerns or critiques are expressed in a productive manner. In this situation the people who concerns it have to follow the following steps: <ul style="list-style-type: none"> <li>• situational information to describe what is happening;</li> <li>• background information or context;</li> <li>• an assessment of what they believe the problem is;</li> <li>• recommendations for how to proceed.</li> </ul>
Security testing	Security testing is one of many types of tests. Within DevOps security testing is integrated in the deployment pipeline by using automated tests as early as possible in the flow.
Self service capability	One way of integrating Ops in Dev is the usage of infrastructure self-services.
Shared goals	Delivering value to the customer requires that Dev and Ops are working together in value streams and have shared goals and practices.
Shared Operations Team (SOT)	A SOT is a team that is responsible for managing all the DTAP environments performing daily deployments into those development and test environments, as well as doing periodically production deployments. The reason to use a SOT is to have a team that focusses only on deployments. This results in automation of repeatable work and learning how to fix occurring problems very fast.
Shared version control repository	In order to be able to use trunk-based development DevOps engineers need to share their source code. The source code must be committed into a <u>single repository</u> that also supports version control. Such a repository is called a shared version control repository.
Simian army	Simian Army consists of services (Monkeys) for generating various kinds of failures, detecting abnormal conditions, and testing the ability to survive them. The goal is to keep the cloud service safe, secure, and highly available. Currently there are 3 Monkeys in the Simian Army: <ul style="list-style-type: none"> <li>• Janitor Monkey (unused resources);</li> <li>• Chaos Monkey (try to shut down a service);</li> <li>• Conformity Monkey (non-conformance to rules).</li> </ul>
Single repository	A single repository is used to facilitate trunk-based development.
Smoke testing	Smoke testing is one of the test types that is used to determine whether or not the basics of a new or adjusted service works. Only a few testcases are needed to indicate whether or not at least the most important functions are working properly. This test type origins from the hardware manufacturers where engineers tested circuits by powering on the system and checking for smoke which was an alarm of malfunctioning hardware.
Standard deviation	In statistics, the standard deviation (SD, also represented by the Greek letter sigma $\sigma$ or the Latin letter s) is a measure that is used to quantify the amount of variation or dispersion of a set of data values. A low standard deviation indicates that the data points tend to be close to the mean (also called the expected value) of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values <a href="#">[Wiki]</a> .
Standard operations	The standard operations is the situation in which the system performs as designed. Deviations of the standard operations need to be detected as early as possible.



Begrip	Betekenis
Static analysis	Static analysis is a type of testing that is performed in a non-runtime environment, ideally in the deployment pipeline. Typically, a static analysis tool will inspect program code for all possible run-time behaviours and seek out coding flaws, back doors, and potentially malicious code [Kim 2016].
Swarming	<p>David Bernstein explains how swarming helps to build an effective team which is able to focus and solve complex problems: "When swarming, the whole team works together on the same problem. It helps to know each other and work well together. Generally, groups need to go through the phases of forming (getting to know each other) and storming (having conflicts and resolving them) before they get to performing (being a highly functional team), so give everyone the space to become a team."</p> <p>According to Dr. Spear, the goal of swarming is to contain problems before they have a chance to spread, and to diagnose and treat the problem so that it cannot recur. "In doing so," he says, "they build ever-deeper knowledge about how to manage the systems for doing our work, converting inevitable up-front ignorance into knowledge." [Kim 2016].</p>
System of Engagement (SoE)	SoE's are decentralised Information Communication Technology (ICT) components that incorporate communication technologies such as social media to encourage and enable peer interaction [What-is].
System of Information (SoI)	The term SOI includes are all the tools that are used to process and visualise information from SoR systems. Typically, examples are Business Intelligence (BI) systems.
System of Records (SoR)	<p>A SoR is an ISRS (information storage and retrieval system), that is the authoritative source for a particular data element in a system containing multiple sources of the same element.</p> <p>To ensure data integrity, there must be one -- and only one -- system of record for a given piece of information [What-is].</p>
Technology adaption curve	It takes time for new technology to get adapted in the market. The technology adaption curve indicates the stages of market penetration in time.
Technology executives	This is a DevOps role also named 'value stream manager'. The value stream manager is someone who is responsible for "ensuring that the value stream meets or exceeds the customer (and organisational) requirements for the overall value stream, from start to finish" [Kim 2016].
Test Driven Development (TDD)	Test driven development is the approach in which the source code is written after the completion of the test case definition and execution. The source code is written and adjusted until the test case conditions are met.
Test harness	Software constructed to facilitate integration testing. Where test stubs are typically components of the application under development and are replaced by working components as the application is developed (top-down integration testing), test harnesses are external to the application being tested and simulate services or functionality not available in a test environment.
The Agile Manifesto	The Agile Manifesto (Manifesto for Agile Software Development) was set up during an informal meeting of seventeen software DevOps engineers. This meeting took place from 11 to 13 February 2001 at "The Lodge" in Snowbird, Utah.

Begrip	Betekenis
	<p>The charter and the principles formed an elaboration of ideas that had arisen in the mid-nineties, in response to methods traditionally classed as waterfall development models. Those models were experienced as bureaucratic, slow, and narrow-minded and would hinder the creativity and effectiveness of DevOps engineers. The seventeen people who have drawn up the Agile Manifesto together represented the various Agile movements.</p> <p>After the publication of the charter, several signatories set up the "Agile:- Alliance" to further convert the principles into methods <a href="#">[Wiki]</a>.</p>
The ideal testing automation pyramid	<p>The ideal testing automation pyramid is a way of testing that can be characterised as follows:</p> <ul style="list-style-type: none"> <li>• Most of the errors are found using unit tests as early as possible.</li> <li>• Run faster-running automated tests (e.g., unit tests) before slower-running automated tests (e.g., acceptance and integration tests), which are both run before any manual testing.</li> <li>• Any errors should be found with the fastest possible category of testing.</li> </ul>
The Lean movement	<p>An operating philosophy that stresses listening to the customer, tight collaboration between management and production staff, eliminating waste and boosting production flow. Lean is often heralded as manufacturers' best hope for cutting costs and regaining their innovative edge.</p>
The non-ideal testing automation inverted pyramid	<p>The non-ideal testing automation pyramid is a way of testing that can be characterised as follows:</p> <ul style="list-style-type: none"> <li>• Most of the investment is in manual and integration testing.</li> <li>• Errors are found later in the testing.</li> <li>• Slower running automated tests are performed first.</li> </ul>
The Simian Army	<p>The Simian Army is a collection of open-source cloud testing tools created by the online video streaming company, Netflix. The tools allow engineers to test the reliability, security, resiliency and recoverability of the cloud services that Netflix runs on Amazon Web Services (AWS) infrastructure <a href="#">[Whatis]</a>.</p> <p>Within this Simian Army the following monkeys are recognised: Chaos Gorilla, Chaos Kong, Conformity Monkey, Doctor Monkey, Janitor Monkey, Latency Monkey and Security Monkey.</p>
The three ways	<p>The three ways are introduced in 'The Phoenix Project: A Novel About IT, DevOps, And Helping Your Business Win' by Gene Kim, Kevin Behr and George Spafford.</p> <p>The Three Ways are an effective way to frame the processes, procedures and practices of DevOps, as well as the prescriptive steps.</p> <ul style="list-style-type: none"> <li>• The first way – flow understand and increase the flow of work (left to right);</li> <li>• The second way – feedback create short feedback loops that enable continuous improvement (right to left);</li> <li>• The third way – Continuous Experimentation and Learning (continuous learning).</li> </ul>
Theory of constraints	<p>This is a methodology for identifying the most important limiting factor that stands in the way of achieving a goal and then systematically improving that constraint until it is no longer the limiting factor.</p>

Begrip	Betekenis
Tool-assisted code review	This is a review technique where authors and reviewers use specialised tools designed for peer code review or facilities provided by the source code repositories [Kim 2016].
Toyota Kata	Toyota Kata is a management book by Mike Rother. The book explains the Improvement Kata and Coaching Kata, which are a means for making the Continual improvement process as observed at the Toyota Production System teachable [Wiki].
Transformation team	Introducing DevOps requires a defined transformation strategy. Based on their research, Dr. Govindarajan and Dr. Trimble assert that organisations need to create a dedicated transformation team that is able to operate outside of the rest of the organisation that is responsible for daily operations (which they call respectively the "dedicated team" and "performance engine"). The lessons learned from this transformation team can be used to apply in the rest of the organisation.
Value stream	The process required to convert a business hypothesis into a technology-enabled service that delivers value to the customer [Kim 2016].
Value Stream Mapping (VSM)	Value stream mapping is a Lean tool that depicts the flow of information, materials, and work across functional silos with an emphasis on quantifying waste, including time and quality.
Vertical splitting of features	A feature can be splitted into stories. Vertical splitting refers to the result of a feature splitting in which more DevOps teams can work independently on their own stories. Together they realise the feature. See also Horizontal splitting of features.
Virtualised environment	An environment that is based on virtualisation of hardware platforms, storage devices and network resources. In order to create a virtualised environment usually VMware is used.
Visualisation	In computing, virtualisation refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources. Virtualisation began in the 1960s, as a method of logically dividing the system resources provided by mainframe computers between different applications. Since then, the meaning of the term has broadened [Wiki].
Walking skeleton	Walking skeleton means doing the smallest possible amount of work to get all the key elements in place.
Waste	Waste comprises the activities that are performed in the manufacturing process that are not adding value to the customer. Examples in the context of DevOps are: <ul style="list-style-type: none"> <li>• Unnecessary software features.</li> <li>• Communication delays.</li> <li>• Slow application response times.</li> <li>• Overbearing bureaucratic processes.</li> </ul>
Waste reduction	Minimisation of waste at its source is to minimise the quantity required to be treated and disposed of, achieved usually through better product design and/or process management. Also called waste minimisation [Businessdictionary].
WIP limit	This is a Key Performance Indicator (KPI) that is used in the Kanban process to maximise the number of items that has been started but that is not completed. Limiting the amount of WIP is an excellent way to increase throughput in your software development pipeline.

<b>Begrip</b>	<b>Betekenis</b>
Work In Progress (WIP)	Material that has entered the production process but is not yet a finished product. Work in progress (WIP) therefore refers to all materials and partly finished products that are at various stages of the production process.

Tabel B-1, Begrippenlijst.

## Bijlage C, Afkortingen

Afkorting	Betekenis
%C/A	Percent Complete / Accurate
AFM	Autoriteit Financiële Markten
AVG	Algemene Verordening Gegevensbescherming
AWS	Amazon Web Services
BDD	Behavior Driven Development
BI	Business Intelligence
BOK	Body of Knowledge
BSC	Balanced Score Card
BVS	Business Value System
CA	Competitive Advantage
CA	Continuous Auditing
CAB	Change Advisory Board
CAMS	Culture, Automation, Measurement and Sharing
CD	Continuous Deployment
CE	Continuous Everything
CEM	Central Event Monitor
CEMLI	Configuration, Extension, Modification, Localisation, Integration
CEO	Chief Executive Officer
CFO	Chief Finance Officer
CI	Configuration Item
CI	Continuous Integration
CIA	Confidentiality, Integrity & Availability
CIO	Chief Information Officer
CL	Continuous Learning
CM	Continuous Monitoring
CMDB	Configuration Management DataBase
CMMI	Capability Maturity Model Integration
CMS	Configuration Management System
CN	Continuous design
CO	Continuous dOcumentation
CoC	Code of Conduct
CoP	Communities of Practice (CoP)
CP	Continuous Planning
CPU	Central Processing Unit
CR	Competitive Response
CRAMM	CCTA Risk Assessment Method Methodology
CRC	Cyclic Redundancy Check
CS	Continuous aSessment
CT	Continuous Testing
CTO	Chief Technical Officer
CY	Continuous security
DevOps	Development & Operations

Afkorting	Betekenis
DML	Definitive Media Library
DNS	Domain Name System
DoD	Definition of Done
DoR	Definition of Ready
DTAP	Development, Test, Acceptance and Production
DU	Definitional Uncertainty
DVS	Development Value System
E2E	End-to-End
ERD	Entity Relation Diagram
ERP	Enterprise Resource Planning
ESA	Epic Solution Approach
ESB	Enterprise Service Bus
ETL	Extract Transform & Load
EUX	End User eXperience Monitoring
FAT	Functionele AcceptatieTest
FSA	Feature Solution Approach
GAT	Gebruiker AcceptatieTest
GCC	General Computer Controls
GDPR	General Data Protection Regulation
GIT	Global Information Tracker
GSA	Generieke & Specifieke Acceptatiecriteria
GUI	Graphical User Interface
GWT	Given-When-Then
HRM	Human Resource Management
HRR	Hand-off Readiness Review
IaC	Infrastructure as Code
ICT	Information Communication Technology
ID	Identifier
INVEST	Independent, Negotiable, Valuable, Estimatable, Small and Testable
IPOPS	Information assets, People, Organisation, Products and services, Systems and processes
IR	Infrastructure Risk
ISAE	International Standard On Assurance Engagements
ISMS	Information Security Management System
ISO	Information Standardisation Organisation
ISVS	Information Security Value System
IT	Information Technology
ITIL 4	Information Technology Infrastructure Library 4
ITSM	Information Technology Service Management
JIT	Just In Time
JKK	Ji-Kotei-Kanketsu
JVM	Java Virtual Machine
KPI	Key Performance Indicator
KSF	Kritieke Succes Factor

Afkorting	Betekenis
LAN	Local Area Network
LCM	LifeCycle Management
LDAP	Lightweight Directory Access Protocol
LRR	Launch Readiness Review
LT	Lead Time
MASR	Modify, Avoid, Share, Retain
MFA	Multi Factor Authentication
MI	Management Information
MOF	Microsoft Operations Framework
MRI	Minimum Required Information
MT	Module Test
MTBF	Mean Time Between Failure
MTBSI	Mean Time Between System Incidents
MTTR	Mean Time To Repair
MVP	Minimal Viable Product
NC	Non Conformity
NFR	Non-Functional Requirement
OAWOW	One Agile Way of Working
OLA	Operational Level Agreement
OTAP	Ontwikkel-, Test-, Acceptatie- en Productieomgeving
PAAS	Platform As A Service
PAT	Productie AcceptatieTest
PBI	Productie Backlog Item
PDCA	Plan Do Check Act
PESTLE	Political, Economic, Sociological, Technological, Legislative, Environmental
POR	Project or Organisational Risk
PPT	People, Process & Technology
PST	Performance StressTest
PT	Processing Time
QA	Quality Assurance
QC	Quality Control
RACI	Responsibility, Accountable, Consulted and Informed
RASCI	Responsibility, Accountable, Supporting, Consulted and Informed
RBAC	Role-Based Access Control
REST API	REpresentational State Transfer Application Programming Interface
ROI	Return On Investment
RUM	Real User Monitoring
S-CI	Software Configuration Item
SA	Strategic IS Architecture
SAFe	Scaled Agile Framework
SAT	Security AcceptatieTest
SBAR	Situation, Background, Assessment, Recommendation

Afkorting	Betekenis
SBB	System Building Block
SBB-A	System Building Block Application
SBB-I	System Building Block Information
SBB-T	System Building Block Technology
SIT	Systeemintegratietest
SLA	Service Level Agreement
SM	Strategic Match
SMART	Specific, Measurable, Accountable, Realistic, Timely
SME	Subject Matter Expert
SNMP	Simple Network Management Protocol
SoA	Statement of Applicability
SoE	System of Engagement
SoI	Systems of Information
SoR	System of Records
SoX	Sarbanes Oxley
SQL	Structured Query Language
SRG	Standards Rules & Guidelines
SSL	Secure Sockets Layer
ST	Systeemtest
SVS	Service Value System
TCO	Total Cost of Ownership
TCP	Transmission Control Protocol
TDD	Test Driven Development
TFS	Team Foundation Server
TISO	Technical Information Security Officer
TOM	Target Operating Model
TPS	Toyota Production System
TTM	Time To Market
TU	Technical Uncertainty
TVB	Taken, Verantwoordelijkheden en Bevoegdheden
UAT	User Acceptance Test
UML	Unified Modeling Language
UT	Unit Testing
UX design	User eXperience design
VCR	Verwachte Contante ROI
VOIP	Voice over Internet Protocol
VSM	Value Stream Mapping
WAN	Wide Area Network
WIP	Work In Progress
WMI	Windows Management Instrumentation
WoW	Way of Working
XML	eXtensible Markup Language
XP	eXtreme Programming

Tabel C-1, Afkortingen.



## Bijlage D, Websites

bigpanda	[Bigpanda]	<a href="https://www.bigpanda.io/blog/event-correlation/">https://www.bigpanda.io/blog/event-correlation/</a>
Bullseye	[Bullseye]	<a href="https://www.bullseye.com/minimum.html">https://www.bullseye.com/minimum.html</a>
Businessdictionary	[Businessdictionary]	<a href="http://www.businessdictionary.com">http://www.businessdictionary.com</a>
Collabnet	[CollabNet]	<a href="https://www.collab.net">https://www.collab.net</a>
CleanArchitecture	[CleanArchitecture]	<a href="https://www.freecodecamp.org/news/a-quick-introduction-to-clean-architecture-990c014448d2/">https://www.freecodecamp.org/news/a-quick-introduction-to-clean-architecture-990c014448d2/</a>
CleanCode	[CleanCode]	<a href="https://cvuorinen.net/2014/04/what-is-clean-code-and-why-should-you-care/">https://cvuorinen.net/2014/04/what-is-clean-code-and-why-should-you-care/</a>
dbmetrics	[dbmetrics]	<a href="http://www.dbmetrics.nl">http://www.dbmetrics.nl</a>
dbmetrics	[dbmetrics publicaties]	<a href="https://www.dbmetrics.nl/wp-content/uploads/2021/07/dbmetrics_best-practice-publicaties_2021-07-22_900.pdf">https://www.dbmetrics.nl/wp-content/uploads/2021/07/dbmetrics_best-practice-publicaties_2021-07-22_900.pdf</a>
De Caluwé	[De Caluwé]	<a href="https://www.agile4all.nl/het-kleurenmodel-van-de-caluwe-en-vermaak/">https://www.agile4all.nl/het-kleurenmodel-van-de-caluwe-en-vermaak/</a>
DevOps	[DevOps]	<a href="http://DevOps.com">http://DevOps.com</a>
DDD	[DDD]	<a href="https://www.slideshare.net/skillsmatter/ddd-in-agile">https://www.slideshare.net/skillsmatter/ddd-in-agile</a>
doxygen	[doxygen]	<a href="http://www.doxygen.nl/manual/docblocks.html">http://www.doxygen.nl/manual/docblocks.html</a>
doxygen voorbeeld	[doxygen voorbeeld]	<a href="http://www.doxygen.nl/manual/examples/qtstyle/html/class_q_tstyle_test.html#a0525f798cda415a94fedeceb806d2c49">http://www.doxygen.nl/manual/examples/qtstyle/html/class_q_tstyle_test.html#a0525f798cda415a94fedeceb806d2c49</a>
EXIN	[Exin]	<a href="http://www.exin.nl">http://www.exin.nl</a>
Gladwell	[GLADWELL]	<a href="http://www.gladwill.nl">http://www.gladwill.nl</a>
IIR	[IIR]	<a href="http://www.IIR.nl">http://www.IIR.nl</a>
Investopedia	[Investopedia]	<a href="https://www.investopedia.com">https://www.investopedia.com</a>
ITMG	[ITMG]	<a href="http://www.ITMG.nl">http://www.ITMG.nl</a>
ITPedia	[ITPEDIA]	<a href="http://www.itpedia.nl">http://www.itpedia.nl</a>
Patrick Cousot	[Patrick Cousot]	<a href="https://www.di.ens.fr/~cousot/abstract_interpret.shtml">https://www.di.ens.fr/~cousot/abstract_interpret.shtml</a>
Porter	[Porter]	<a href="https://medium.com/@sniloy/value-chain-analysis-value-stream-mapping-and-business-process-mapping-what-is-the-difference-431589d27ea8">https://medium.com/@sniloy/value-chain-analysis-value-stream-mapping-and-business-process-mapping-what-is-the-difference-431589d27ea8</a>
Sneider	[Schneider]	<a href="https://shift314.com/are-you-using-the-right-culture-model/">https://shift314.com/are-you-using-the-right-culture-model/</a>
Tiobe	[Tiobe]	<a href="http://www.tiobe.com/content/paperinfo/DefinitionOfConfidenceFactor.html">www.tiobe.com/content/paperinfo/DefinitionOfConfidenceFactor.html</a>
UnitTest	[UnitTest]	<a href="https://docs.python.org/3/library/unittest.html">https://docs.python.org/3/library/unittest.html</a>
Westrum	[Westrum]	<a href="https://www.delta-n.nl/het-belang-van-cultuur-in-devops/">https://www.delta-n.nl/het-belang-van-cultuur-in-devops/</a>
Wiki	[Wiki]	<a href="http://nl.wikipedia.org/wiki/Cloud_computing">http://nl.wikipedia.org/wiki/Cloud_computing</a>
Wiki docgen	[Wiki docgen]	<a href="https://en.wikipedia.org/wiki/Comparison_of_documentation_generators">https://en.wikipedia.org/wiki/Comparison_of_documentation_generators</a>

Tabel D-1, Websites.

## Bijlage E, Index

---

### %

%C/A · 123

---

### A

A/B testing · 94, 107  
 acceptatiecriterium · 108, 112  
 acceptatietest · 107  
 account data · 54  
 actor · 30, 31, 32, 44, 63, 64, 65  
 affinity · 107  
 AFM · 123  
 Agile · 5, 17, 95, 107, 119, 120  
   - coach · 1  
   - design · 3  
   - design pyramid · 77  
   - infrastructure · 107  
   - proces · 95  
   - release trains · 17  
   - Scrum · 16  
   - test management · 43  
 Agile Scrum · 111  
 Agile Scrum framework · 16, 17  
 Agile Scrum proces · 16  
 Algemene Verordening  
   Gegevensbescherming · Zie AVG  
 alignment · 89  
 alternate path · 107  
 Amazon Web Services · Zie AWS  
 Andon cord · 107  
 anker · 13  
 anomaly detection technique · 107  
 antifragility · 108  
 anti-pattern · 7, 14, 15, 18, 19, 20, 21,  
   108  
 apenrots · 14  
 applicatie · 94  
 applicatiebeheer · 114  
 applicatiecomponent · 9, 44, 115  
 application business rule · 80  
 architectuur · 8, 49, 51, 80, 85, 87, 89, 92  
 architectuurprincipe · 87  
 artefact · 29, 108, 110  
 artefact repository · 108  
 assessment · 12, 31, 32, 118  
 auditing · 94  
 auditor · 1  
 automated test · 108  
 Autoriteit Financiële Markten · Zie AFM  
 availability · 109  
 AVG · 2, 123  
 AWS · 54, 123  
 Azure · 54

---

### B

backlog item · 117  
 bad apple theory · 108  
 bad path · 108  
 Balanced Score Card · Zie BSC  
 base use case · 31  
 baseline · 94  
 BDD · 4, 28, 69, 70, 71, 73, 74, 78, 79,  
   80, 91, 94, 108, 123  
 BDD-aanpak · 28  
 bedrijfsproces · 35, 94  
 beeldvorming · 4, 13, 14, 15, 18, 20  
 Behavior Driven Development · Zie BDD  
 beleid · 56  
 best practice · 13, 96, 109  
 besturing' · 7  
 beveiliging · 95  
 bevoegdheden · 4, 13  
 BI · 123  
 BI-model · 2  
 binary · 108  
 biometrieke · 56  
 BI-oplossing · 3  
 black-box · 56  
 blameless post mortem · 94, 108  
 blamelessness · 108  
 blue/green deployment · 108  
 blue/green environment · 94  
 Body of Knowledge · Zie BOK  
 BOK · 123  
 bottom-up · 13  
 bouwsteen · 9, 43, 47, 48, 49, 50, 51, 52,  
   53, 56, 58, 59, 63, 67  
 branching · 94, 109  
 broken build · 94, 108  
 brown field · 108  
 BSC · 123  
 build · 94, 108, 109, 110, 119  
 build-in failure mode · 94  
 business  
   - architect · 37  
   - DevOps · 94  
   - information manager · 37  
   - transactie · 54  
   - value · 109, 111  
   - value stream · 18  
   - view · 4, 7, 35  
 Business Canvas Model · 40  
 Business Intelligence · Zie BI  
 Business Value System · Zie BVS  
 BVS · 123

---

### C

C/A · 39  
 CA · 123

- CAB · 89, 123
- CAMS · 110, 123
- canary releasing · 94, 109
- capability · 110
- Capability Maturity Model Integration · Zie CMMI
- capaciteit · 109
- CCTA Risk Assessment Method Methodology · Zie CRAMM
- CD · 94, 96, 109, 113, 123
- CE · 96, 123
- CEM · 123
- CEMLI · 123
- CE-model · 93
- Central Event Monitor · Zie CEM
- Central Processing Unit · Zie CPU
- CEO · 123
- certificaat · 54
- CFO · 123
- Change Advisory Board · Zie CAB
- change category · 109
- change schedule · 109
- check-in · 94
- checkpunt · 98
- Chief Executive Officer · Zie CEO
- Chief Finance Officer · Zie CFO
- Chief Information Officer · Zie CIO
- Chief Technology Officer · Zie CTO
- CI · 93, 94, 96, 109, 113, 123
- CI/CD secure pipeline · 2, 16, 20
- CIA · 123
- CIO · 123
- CL · 94, 123
- cloud · 109
  - broker · 56
  - configuration file · 109
  - provider · 56
  - service · 109
- cluster immune system release pattern · 109
- CM · 94, 97, 98, 123
- CMDB · 123
- CMMI · 93, 95, 96, 97, 123
- CMS · 123
- CN · 123
- CO · 123
- CoC · 123
- code branch · 109
- Code of Conduct · Zie CoC
- code review form · 109
- code view · 4, 7
- codified NFR · 109
- collaboratie tool · 21, 37, 64
- collaboration · 109
- collaboration tool · 37
- commandline · 75
- commit code · 109
- commit stage · 109
- communication service · 56
- Communities of Practice · Zie CoP
- competence · 107, 112
- Competitive Advantage · Zie CA
- Competitive Response · Zie CR
- Completeness / Accurateness · Zie %C/A
- compliance · 94, 110
- compliance checking · 110
- compliance officer · 110
- compliance officer · 110
- component · 113, 116, 119
- confidentiality · 123
- Confidentiality, Integrity & Availability · Zie CIA
- configuratiebeheer · 94
- Configuration Item · Zie CI
- configuration management · 110
- Configuration Management DataBase · Zie CMDB
- Configuration Management System · Zie CMS
- Configuration, Extention, Modification, Localisation, Integration · Zie CEMLI
- Confluence · 19, 37, 64
- container · 110
- continuity · 109, 115
- Continuous
  - Assessment · 2, 5, 11, 12, 14, 15, 16, 25, 27, 30, 31
  - Auditing · 2
  - Delivery · 96
  - Deployment · 2
  - Design · 2
  - Documentation · 2
  - Everything · 1, 11, 93, 94, 95, 96
  - Improvement · 120
  - Integration · 2
  - Learning · 2, 120
  - Monitoring · 2
  - Planning · 2
  - Testing · 2
  - Testing roadmap · 12, 17
- Continuous aSessment · Zie CS
- Continuous Auditing · Zie CA
- Continuous Deployment · Zie CD
- Continuous desigN · Zie CN
- Continuous Design pyramid · 7, 13, 19, 25, 27, 28, 35, 43, 51, 61, 69, 73, 77, 85
- continuous documentation · 4, 77, 78, 79, 80
- Continuous dOcumentation · Zie CO
- Continuous Everything · Zie CE
- Continuous Integration · Zie CI
- Continuous Learning · Zie CL
- Continuous Monitoring · Zie CM
- Continuous Planning · Zie CP
- Continuous securitY · Zie CY
- Continuous Testing · Zie CT
- control · 96, 110, 118
- Conway's law · 110
- CoP · 17, 123
- counter measure · 114
- CP · 123
- CPU · 123
- CR · 123
- CRAMM · 123
- CRC · 123
- CS · 123

CT · 94, 123  
 CTO · 123  
 Cucumber · 70, 74  
 cultural debt · 110  
 Culture, Automation, Measurement and Sharing · Zie CAMS  
 current state · 38, 39, 40  
 CY · 123  
 cycle time · 94, 110  
 Cyclic Redundancy Check · Zie CRC

---

## D

data · 96  
 data storage · 56  
 data storage service · 56  
 debt · 110  
 declarative programming · 110  
 defect · 95, 115  
 defect tracking · 111  
 Definition of Done · Zie DoD  
 Definition of Ready · Zie DoR  
 Definitional Uncertainty · Zie DU  
 Definitive Media Library · Zie DML  
 deliverable · 4, 14, 21, 28, 35, 43, 61, 69, 73, 77  
 Demming wheel · 114  
 deployment · 107  
 deployment pipeline · 4, 52, 54, 55, 58, 59, 64, 73, 74, 77, 78, 79, 109  
 deployment pipeline tool · 54  
 design · 2, 110, 121  
 design view · 4, 7, 63  
 Dev engineer · 1  
 development · 107, 108, 111, 113, 114, 116, 118, 119, 120, 121  
 Development & Operations · Zie DevOps  
 development ritual · 111  
 development team · 16, 17  
 Development Value System · Zie DVS  
 Development, Test, Acceptance and Production · Zie DTAP  
 DevOps · 3, 13, 14, 16, 93, 95, 96, 103, 107, 109, 111, 117, 121, 123  
 - capabilities · 12  
 - coach · 87  
 - engineer · 18, 107, 109, 110, 111, 118, 119, 120  
 - Lemniscaat · 1, 2  
 - team · 1, 16, 17, 89, 107, 108, 109, 110, 112, 116, 121  
 - werkwijze · 1  
 DML · 124  
 DMZ service · 55  
 DNS · 124  
 DNS service · 55  
 documentatie management · 54  
 documentatiegenerator · 77, 78, 81  
 DoD · 2, 9, 31, 32, 74, 79, 95, 113, 124  
 Domain Name System · Zie DNS  
 DoR · 95, 124  
 downward spiral · 111

DTAP · 118, 124  
 DTAP environments · 118  
 DU · 124  
 DVS · 124

---

## E

E2E · 95, 124  
 effectief · 16  
 efficiënt · 16  
 eigenaarschap · 12, 13, 16, 18  
 e-mail pass around · 94, 111  
 End User eXperience Monitoring · Zie EUX  
 End-to-End · Zie E2E  
 enterprise architectuur · 9, 59  
 enterprise business rule · 80  
 Enterprise Resource Planning · Zie ERP  
 Enterprise Service Bus · Zie ESB  
 Entity Relation Diagram · Zie ERD  
 epic · 14, 21, 28, 40, 43, 62, 69, 95, 124  
 Epic one pager · 33  
 Epic Solution Approach · Zie ESA  
 epic-ontwerp · 90  
 ERD · 124  
 ERP · 124  
 error path · 111  
 ESA · 87, 89, 124  
 ESB · 124  
 E-shaped · 94, 112  
 ETL · 124  
 EUX · 124  
 event · 94, 116  
 eXtensible Markup Language · Zie XML  
 Extract Transform & Load · Zie ETL  
 eXtreme Programming · Zie XP

---

## F

failure · 108  
 fast feedback · 13, 85  
 FAT · 107, 124  
 feature · 21, 28, 40, 62, 64, 69, 70, 71, 73, 78, 79, 81, 87, 91, 95, 111, 112, 121  
 Feature Solution Approach · Zie FSA  
 feature toggle · 94, 111  
 feature-ontwerp · 90  
 feedback · 14, 22, 23, 47, 86, 94, 95, 110, 111, 114, 117, 120  
 feedforward · 111  
 first time right · 39  
 five times why methode · 12  
 flow · 8, 35, 36, 37, 40, 44, 68, 81, 94, 95, 110, 113, 114, 115, 116, 117, 118, 120, 121  
 framework · 116  
 front-end applicatie · 16  
 FSA · 87, 88, 89, 90, 91, 92, 124  
 functie · 1, 17  
 functioneel ontwerp · 8  
 Functionele AcceptatieTest · Zie FAT

**G**

Galbraith · 17  
 GAT · 124  
 Gaussian distribution · 107, 111  
 GCC · 2, 124  
 GDPR · 124  
 Gebruiker AcceptatieTest · Zie GAT  
 Gene Kim · 111, 116, 120  
 General Computer Controls · Zie GCC  
 General Data Protection Regulation · Zie GDPR  
 Generieke & Specifieke Acceptatiecriteria · Zie GSA  
 gereedschap · 94  
 gewenste gedrag · 11  
 Gherkin · 19, 70, 71, 73, 74, 78, 79  
 Gherkin language · 9, 27, 70, 71, 74, 79  
 Gherkin taal · 19  
 Git · 90  
 GIT · 124  
 Given When Then · 112, Zie GWT  
 Global Information Tracker · Zie GIT  
 golden plated · 21  
 governance · 17  
 Graphical User Interface · Zie GUI  
 green build · 94  
 green field · 112  
 GSA · 124  
 GUI · 124  
 GWT · 9, 28, 32, 33, 70, 74, 81, 112, 124

**H**

Hand-off Readiness Review · Zie HRR  
 happy flow · 29  
 happy path · 107, 108, 112  
 hardware · 110, 113, 121  
 hergebruik · 74  
 Holocracy · 112  
 horizontal splitting of feature · 112, 121  
 HRM · 2, 25, 124  
 HRR · 94, 124  
 Human Resource Management · Zie HRM  
 hypothesis driven development · 94

**I**

IaC · 4, 78, 107, 113, 124  
 ICT · 113, 124  
 ID · 124  
 ideal test pyramid · 94, 120  
 idempotent · 112  
 IDentifier · Zie ID  
 impact · 3  
 impactanalyse · 37  
 imparative programming · 112  
 Independent, Negotiable, Valuable, Estimatable, Small and Testable · Zie INVEST

informatiesysteem · 16  
 Information assets, People, Organisation, Products and services, Systems and processes · Zie IPOPS  
 Information Communication Technology · Zie ICT  
 information radiator · 113  
 Information Security Management System · Zie ISMS  
 Information Security Value System · Zie ISVS  
 Information Standardisation Organisation · Zie ISO  
 Information Technology · Zie IT  
 Information Technology Infrastructure Library · Zie ITIL 4  
 Information Technology Service Management · Zie ITSM  
 Infosec · 113  
 Infrastructure as Code · Zie IaC  
 infrastructure component · 113  
 infrastructure management · 113  
 Infrastructure Risk · Zie IR  
 infrastuctuur monitortool · 55  
 integrity · 123  
 intermediair · 85  
 International Standard On Assurance Engagements · Zie ISAE  
 INVEST · 112, 124  
 IP address · 113  
 IPOPS · 124  
 IR · 124  
 I-shaped · 94, 112  
 ISMS · 124  
 ISO · 124  
 IST · 114  
 ist situatie · 86  
 ISVS · 124  
 IT · IV, 2, 28, 49, 51, 111, 115, 120, 124  
 ITIL 4 · 124  
 ITSM · 115, 124

**J**

Java Virtual Machine · Zie JVM  
 Ji-Kotei-Kanketsu · Zie, Zie JKK  
 JIT · 113, 114, 124  
 JKK · 113, 124  
 JSON · 78  
 Just In Time · Zie JIT  
 JVM · 124

**K**

Kaizen · 113, 115  
 Kaizen Blitz (or Improvement Blitz) · 114  
 Kaizen in advance · 114  
 Kanban · 114, 115, 121  
 Ken Schwaber · 16  
 kennisoverdracht · 2, 12  
 keten · 16, 94

Key Performance Indicator · Zie KPI  
 kibana dashboard · 114  
 kleurenmodel · 127  
 KPI · 114, 115, 121, 124  
 Kritieke Succes Factor · Zie KSF  
 KSF · 124  
 kwaliteit · 17  
 kwaliteitscriteria · 95

---

## L

lagenstructuur · 28  
 LAN · 125  
 latent defect · 114  
 Launch Readiness Review · Zie LRR  
 launching guidance · 114  
 LCM · 125  
 LDAP · 125  
 LDAP service · 55  
 Lead Time · 114, Zie LT  
 Lean · 120, 121  
 Lean Six Sigma · 9, 39, 59  
 Lean tool · 114  
 learning culture · 115  
 levenscyclus · 26  
 lifecycle · 111, 113  
 LifeCycle Management · Zie LCM  
 Lightweight Directory Access Protocol · Zie LDAP  
 lijnmanager · 1  
 Local Area Network · Zie LAN  
 log · 94, 116  
 logging level · 115  
 loosely coupled architecture · 115  
 loosely coupled services · 115  
 LRR · 94, 114, 125  
 LT · 39, 59, 65, 114, 125

---

## M

machtsverhouding · 4, 13, 15, 16, 18, 20  
 management · 96  
 Management Information · Zie MI  
 manufacturing process · 121  
 marker · 9, 32, 33  
 marktsegment · 40  
 MASR · 125  
 Mean Time Between Failure · Zie MTBF  
 Mean Time Between System Incidents · Zie MTBSI  
 Mean Time To Repair · Zie MTTR  
 merging · 94  
 meta-data · 108  
 metadata · 14, 63, 80  
 methodology · 96  
 metrics · 94  
 MFA · 125  
 MI · 125  
 Michael Porter · 104  
 microservice · 77, 79, 115  
 microservice architecture · 115

Microsoft Operations Framework · Zie MOF  
 mini pipeline · 115  
 Minimal Viable Product · Zie MVP  
 Minimum Required Information · Zie MRI  
 Modify, Avoid, Share, Retain · Zie MASR  
 modulair programmeren · 79  
 Module Test · Zie MT  
 MOF · 125  
 monitor  
 - architecture · 94  
 monitoring · 116  
 monolithic · 116  
 MRI · 115, 125  
 MT · 125  
 MTBF · 125  
 MTBSI · 125  
 MTTR · 116, 125  
 muda · 116  
 Multi Factor Authentication · Zie MFA  
 MVP · 43, 125

---

## N

NC · 125  
 netwerkcommunicatie · 55  
 NFR · 58, 109, 116, 125  
 non · 8  
 Non Conformity · Zie NC  
 Non Functional Requirement · Zie NFR  
 non-SQL database · 54  
 nulmeting · 7

---

## O

OAWOW · 125  
 obeya · 116  
 object code · 108  
 Oirsouw · 105  
 OLA · 125  
 One Agile Way of Working · Zie OAWOW  
 one piece flow · 116  
 one-delivery-pipeline · 56  
 ontwerp · 89, 90  
 ontwerptechniek · 19  
 Ontwikkel-, Test-, Acceptatie- en Productieomgeving · Zie OTAP  
 ontwikkelproces · 16, 80  
 onzekerheidsreductie principe · 17  
 Operational Level Agreement · Zie OLA  
 operations · 107, 111, 116, 118, 121  
 operations story · 116  
 Ops engineer · 1  
 Ops liaison · 116  
 organisatievormgeving · 4  
 organisation archetype · 116  
 organisational typology model · 116  
 OTAP · 125  
 output stroom · 8, 35, 36, 44  
 over-the-shoulder · 94, 117

**P**

PAAS · 56, 125  
 package · 117  
 pair programming · 94, 109, 117  
 PAT · 107, 125  
 patch · 3  
 pattern · 88, 89, 90, 108, 117  
 PBI · 125  
 PDCA · 114, 115, 125  
 peer review · 117  
 peer to peer programming · 109  
 People, Process & Technology · Zie PPT  
 performance · 109, 115, 121, 125  
 Performance StressTest · Zie PST  
 performancecriterium · 30  
 personalisatie · 54  
 PESTLE · 125  
 pipeline · 94, 95, 107, 113, 115, 118, 119, 121  
 Plan Do Check Act · Zie PDCA  
 Platform As A Service · Zie PAAS  
 Political, Economic, Sociological, Technological, Legislative, Environmental · Zie PESTLE  
 POR · 125  
 post mortem · 117  
 post-condition · 63  
 PPT · 2, 11, 25, 49, 125  
 PPT-aspect · 94  
 pre-condition · 63, 70  
 prestatie indicatoren · 59  
 principe · 14  
 probleem · 11, 12  
 proceseigenaar · 1  
 procesmanager · 1  
 Processing Time · Zie PT  
 product
 

- backlog · 14, 15, 17, 28, 54, 62, 117
- backlog item · 17, 113
- owner · 1, 117

 Product Backlog Item · Zie PBI  
 Productie AcceptatieTest · Zie PAT  
 productieomgeving · 80  
 production environment · 115  
 programming paradigm · 117  
 Project or Organisational Risk · Zie POR  
 PSQL · 110  
 PST · 125  
 PT · 39, 59, 65, 125  
 pull request · 80  
 pull request process · 94, 117  
 Python · 74, 75, 76, 80, 81

**Q**

QA · 95, 117, 125  
 QC · 125  
 quality · 96  
 Quality Assurance · Zie QA  
 Quality Control · Zie QC

**R**

RACI · 17, 125  
 RASCI · 17, 125  
 RBAC · 125  
 RBAC service · 55  
 Real User Monitoring · Zie RUM  
 reduce batch size · 117  
 reduce number of handoffs · 117  
 redundantie · 15  
 referentie-architectuur · 90  
 refinement · 14  
 refinieren · 62  
 release · 2, 3, 117  
 release manager · 117  
 release pattern · 117  
 repository · 94, 95, 108, 109, 117, 118  
 REpresentational State Transfer Application Programming Interface · Zie REST API  
 requirement · 4, 8, 15, 19, 21, 28, 44, 56, 61, 64, 69, 70, 71, 73, 77, 78, 80, 81, 90, 91, 94, 108, 114, 116, 119, 125  
 requirement view · 4, 7  
 research · 1  
 resource · 4, 14  
 Responsibility, Accountable, Consulted and Informed · Zie RACI  
 Responsibility, Accountable, Supporting, Consulted and Informed · Zie RASCI  
 REST API · 125  
 retrospective · 111  
 Return On Investment · Zie ROI  
 review · 111  
 risico · 1, 3, 87, 95, 108, 114  
 risicoanalyse · 37  
 roadmap · 13  
 ROI · 125  
 rol · 17, 29  
 Role-Based Access Control · Zie RBAC  
 rollback technique · 94  
 rootcause · 11  
 rootcause analyse · 115  
 RUM · 125

**S**

SA · 125  
 sad path · 117  
 SAFe · 125  
 SAFe framework · 17, 18  
 safety check · 117  
 Sarbanes Oxley · Zie SoX  
 SAT · 125  
 SBAR · 118, 125  
 SBB · 31, 126  
 SBB-A · 126  
 SBB-I · 48, 49, 50, 51, 53, 54, 56, 60, 64, 65, 66, 72, 126  
 SBB-I diagram · 48, 49, 50, 53, 54  
 SBB-T · 31, 48, 54, 56, 60, 64, 67, 126  
 Scaled Agile Framework · Zie SAFe

S-CI · 125  
 Scrum master · 1, 16, 17  
 Secure Sockets Layer · Zie SSL  
 security · 109, 110, 116, 117, 118, 120  
 Security Acceptatie Test · Zie SAT  
 security officer · 110  
 self service capability · 118  
 service · 125  
 Service Level Agreement · Zie SLA  
 service management · 2  
 service request · 86  
 Service Value System · Zie SVS  
 shared goals · 118  
 shift left · 13  
 shift-left organisatie · 74, 86  
 silo · 121  
 Simian army · 94, 118, 120  
 Simple Network Management Protocol · Zie SNMP  
 SIT · 126  
 Situation, Background, Assessment, Recommendation · Zie SBAR  
 skill matrix · 20, 21  
 skills · 20, 21, 25, 112  
 SLA · 95, 103, 126, 142  
 SM · 126  
 SMART · 115, 126  
 SME · 17, 36, 126  
 smoke testing · 118  
 SNMP · 126  
 SoA · 126  
 SoE · 2, 3, 16, 119, 126  
 software · 15, 96, 108, 119, 121  
 Software Configuration Item · Zie S-CI  
 software item · 61  
 software ontwikkelproces · 69  
 softwareontwikkelproces · 96  
 SoI · 2, 119, 126  
 SOLL · 114  
 soll situatie · 86  
 solution view · 4, 7, 43  
 solution View · 35  
 SoR · 2, 3, 16, 21, 54, 119, 126  
 sourcecode · 4, 8, 9, 46, 73, 74, 75, 77, 78, 79, 80, 81, 89, 91, 94, 108, 109, 111, 118, 119, 121  
 SoX · 126  
 SoX-wetgeving · 2  
 Specific, Measurable, Accountable, Realistic, Timely · Zie SMART  
 Spotify · 17  
 Spotify model · 17, 86  
 sprint · 111  
 sprint execution · 111  
 sprint planning · 111  
 SQL · 126  
 SRG · 94, 95, 126  
 SSL · 126  
 ST · 126  
 stakeholder · 2, 4, 8, 9, 11, 25, 28, 32, 35, 36, 37, 38, 40, 44, 45, 61, 70, 78, 87, 89, 112  
 standard deviation · 118

standard operations · 118  
 Standard Rules & Guidelines · Zie SRG  
 stand-up · 111  
 Statement of Applicability · Zie SoA  
 static analysis · 119  
 story · 21, 40, 69  
 Strategic IS Architecture · Zie SA  
 Strategic Match · Zie SM  
 strategy · 96  
 Structured Query Language · Zie SQL  
 Subject Matter Expert · Zie SME  
 super use case · 31  
 SVS · 126  
 systeemontwikkeling · 2  
 system building block · 43  
 System Building Block · Zie SBB  
 System Building Block Application · Zie SBB-A  
 System Building Block Infrastructure · Zie SBB-I  
 System Building Block Technology · Zie SBB-T  
 System Building Blocks · 4, 31, 43, 44, 47, 48, 58, 64  
 system context diagram · 4, 35, 36, 37, 38, 40, 44  
 System Context Diagram · 8, 31, 32  
 System Integration Test · Zie SIT  
 System of Engagement · Zie SoE  
 System of Records · Zie SoR  
 System Test · Zie ST  
 Systems of Information · Zie SoI

---

## T

taak · 2, 59, 62, 107  
 Taken, Verantwoordelijkheden en Bevoegdheden · Zie TVB  
 Target Operating Model · Zie TOM  
 task · 113  
 TCO · 126  
 TCP · 126  
 TDD · 4, 9, 73, 74, 78, 79, 80, 91, 94, 119, 126  
 Team Foundation Server · Zie TFS  
 technical debt · 13, 96, 110, 111  
 technical debt backlog · 13, 96  
 Technical Information Security Officer · Zie TISO  
 Technical Uncertainty · Zie TU  
 technisch ontwerp · 8, 73  
 technology adaption curve · 119  
 technology executive · 119  
 telemetry · 94  
 template · 30, 31, 93  
 test
 

- case · 4, 23, 56, 61, 64, 73, 74, 75, 107, 108, 109
- framework · 74
- harness · 119
- inspanning · 73, 74
- management · 94



- view · 4, 7
- Test Driven Development · Zie TDD
- testen · 17
- tester · 111
- TFS · 64, 126
- The Agile Manifesto · 119
- the ideal testing automation pyramid · 120
- The Lean movement · 120
- the non-ideal testing automation inverted pyramid · 120
- The Three Ways · 116, 120
- theme · 14, 21, 28, 38, 40, 62, 69, 95
- theory of constraints · 120
- Time To Market · Zie TTM
- tips en trucks · 35, 38, 43, 47, 62, 70, 74, 78
- TISO · 126
- toetsen · 17
- TOM · 126
- tool-assisted code review · 94, 121
- Total Cost of Ownership · Zie TCO
- Toyota Kata · 121
- Toyota Production System · Zie TPS
- TPS · 126
- traceability · 94
- traceerbaarheid · 12, 15, 21, 32, 46, 94, 95
- trage feedback · 11
- transactieverwerking · 16
- transformation team · 121
- Transmission Control Protocol · Zie TCP
- trunk · 118
- T-shaped · 94, 112
- TSQL · 110
- TTM · 126
- TU · 126
- TVB · 126

---

## U

- UAT · 126
- UML · 62, 126
- Unified Modeling Language · Zie UML
- uniformeren · 18
- Unit Test · Zie UT
- Unit testcases · 32
- unittest · 74, 75, 76, 78, 81
- Use Case · 4, 8, 9, 15, 29, 30, 31, 33, 34, 38, 39, 43, 44, 45, 46, 54, 58, 59, 61, 62, 63, 64, 65, 67, 68, 69, 80, 91
- Use Case Diagram · 4, 8, 9, 29, 30, 31, 33, 43, 44, 45, 46, 58, 61, 91
- Use Case Model · 90
- Use Case Narrative · 61, 62, 63, 64, 65, 67, 68
- Use Case Scenario · 61, 62, 63, 67, 68
- User Acceptance Test · Zie UAT
- User eXperience design · 94
- User eXperience design · Zie UX design
- User Story · 3
- User Story · 28
- userinterface · 38, 49

- UT · 126
- UX design · 126

---

## V

- vakspecialisatie · 23
- value stream · 2, 4, 8, 14, 26, 29, 31, 33, 35, 36, 37, 38, 39, 40, 43, 44, 46, 51, 56, 58, 59, 94, 95, 111, 115, 118, 119, 121, 126
- value stream canvas · 4, 8, 35, 36, 37, 38, 40, 44, 46, 51
- Value Stream Canvas · 8, 31, 32, 39, 41
- Value Stream Canvas model · 4, 8, 32, 35, 40, 46, 51
- Value Stream Mapping · 31, 33, Zie VSM
- VCR · 126
- velocity · 108
- veranderparadigma · 4, 13, 14, 15, 18, 20, 85, 87
- verantwoordelijkheden · 4, 13
- versie · 3
- versiebeheer · 32, 37, 51, 64, 69, 74, 75, 77, 90, 94
- vertaalfout · 32
- vertical splitting of feature · 121
- Verwachte Contante ROI · Zie VCR
- verzekeringsmaatschappij · 85
- virtual machine · 55
- virtualised environment · 121
- visibility · 94
- visie · 14
- visualisatie · 121
- Voice over Internet Protocol · Zie VOIP
- VOIP · 126
- volwassenheid · 16, 17, 18, 98
- volwassenheidsmodel · 8
- volwassenheidsniveau · 96
- voortbrengingsproces · 3
- VSM · 4, 43, 58, 59, 95, 121, 126

---

## W

- Wachttijd · 11
- walking skeleton · 121
- WAN · 126
- war room · 116
- waste · 88, 108, 110, 113, 114, 116, 120, 121
- waste reductie · 121
- waterval design* · 15, 21, 22
- watervalaanpak · 21, 22, 23
- watervalproject · 2
- Way of Working · Zie WoW
- webcontent management · 54
- Westrum · 116, 117
- Wide Area Network · Zie WAN
- Windows Management Instrumentation · Zie WMI
- WIP · 126
- WMI · 126

Work In Progress · Zie WIP  
workflow · 110  
WoW · 14, 126

XP · 73, 126

---

**X**

XML · 126

---

**Z**

zoekfaciliteit · 54

## Nawoord

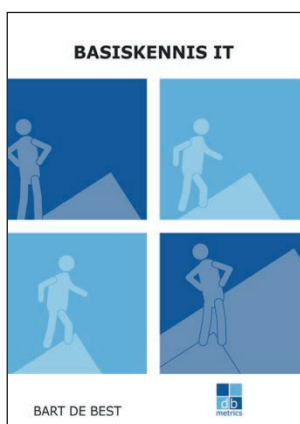
Mijn ervaring is dat de denkbeelden die ik vastleg in een artikel of een boek zich blijven evolueren. In geval u met een bepaald onderwerp uit dit boek aan de slag gaat in uw eigen DevOps organisatie, dan raad ik u aan om even met mij contact op te nemen. Wellicht zijn er aanvullende artikelen of ervaringen op dit gebied die ik met u kan delen. Dit geldt ook omgekeerd evenredig. Als u bepaalde ervaringen hebt die een aanvulling zijn op hetgeen in dit boek is beschreven, dan nodig ik u uit om dit met mij te delen. U kunt mij bereiken via mijn e-mail adres bartb@dbmetrics.nl.

## Over de auteur



**Drs. Ing. B. de Best RI** is vanaf 1985 werkzaam in de ICT. Hij heeft voornamelijk bij de top 100 van het Nederlandse bedrijfsleven en de overheid gewerkt. Hierbij heeft hij gedurende 12 jaar functies vervuld in alle fasen van de systeemontwikkeling, inclusief exploitatie en beheer. Daarna heeft hij zich toegelegd op het service management vakgebied. Momenteel vervult hij als consultant alle aspecten van de kennislevenscyclus van service management, zoals het schrijven en geven van trainingen aan ICT-managers en service managers, het adviseren van beheerorganisaties bij het richting geven aan de beheerorganisatie, de beheerinrichting, het verbeteren van beheerprocessen, het uitbesteden van (delen van) de beheerorganisatie en het reviewen en auditen van beheerorganisaties. Hij is op zowel HTS-niveau als Universitair niveau afgestudeerd op het beheervakgebied.

## Andere boeken van deze auteur



### Basiskennis IT

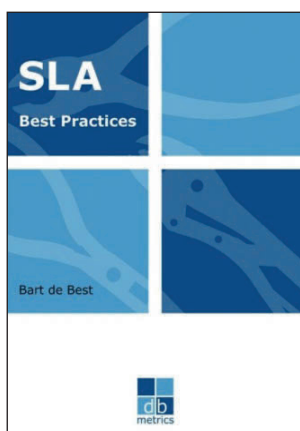
*De eerste stap van een leven lang leren.*

Het boek Basiskennis IT geeft een goede impressie wat dit vakgebied omvat. Zonder dat vele details worden besproken krijgt de lezer een uitleg van de meest essentiële begrippen en concepten van de IT. De doelgroep van dit boek zijn studenten, schoolverlaters en mensen die zich willen laten omscholen tot een beroep in de IT. Daartoe is het een heel nuttig middel als voorbereiding op IT trainingen.

De content bestaat uit het behandelen van IT begrippen uit vier perspectieven te weten het IT landschap, het ontwikkelen van software, het beheren van software en trends in de IT.

Hierbij worden tal van begrippen en concepten behandeld op het gebied van informatie, maatwerkprogrammatuur, systeemprogrammatuur, softwarepakketten, middleware, hardware, netwerk, processen, methoden en technieken. Op deze wijze kunt u snel uw weg vinden in de wereld van IT, het begin van een leven lang leren.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2021  
 ISBN (NL) : 978 94 92618 573



### SLA Best Practices

*Het volledige ABC van service level agreements.*

Het belangrijkste bij het leveren van een service is dat de klant tevreden is over de geleverde prestaties. Door deze tevredenheid verkrijgt de leverancier heraanboren, wordt hij gepromoot in de markt en is de continuïteit van het bedrijf geborgd.

Wellicht nog het belangrijkste aspect van deze klanttevredenheid voor een leverancier is dat de betrokken medewerkers een drive krijgen om hun eigen kennis en kunde verder te ontwikkelen om nog meer klanten tevreden te stellen. Dit boek beschrijft de best practices om erachter te komen wat de Prestatie-Indicatoren (PI's) zijn die gemeten moeten worden om de tevredenheid van de klant te borgen.

Het tweede deel beschrijft de documenten die van toepassing zijn om de afspraken in vast te leggen. Het opstellen, afspreken, bewaken en evalueren van serviceafspraken is een vak op zich. Het derde deel geeft de gereedschappen om hier adequaat invulling aan te geven. De werkzaamheden rond serviceafspraken herhalen zich in de tijd. Deel vier van dit boek beschrijft hoe deze werkzaamheden in een proces gevat kunnen worden en hoe dit proces het beste in een organisatie kan worden vormgegeven. Tot slot geeft bespreekt dit boek een aantal raakvlakken van serviceafspraken en een tweetal artikelen met SLA best practices.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2011  
 ISBN (NL) : 978 90 71501 456



### Cloud SLA

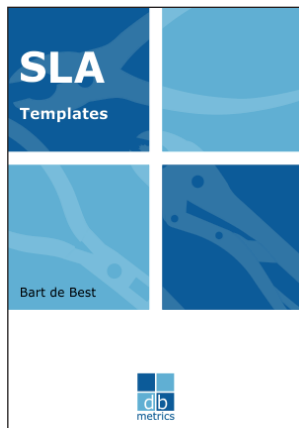
*De best practices van cloud service level agreements*

Steeds meer organisaties kiezen voor het vervangen van de traditionele ICT-services door cloud services. Het opstellen van doelmatige SLA's voor traditionele ICT-services is voor veel organisaties een ware uitdaging. Met de komst van cloud services lijkt dit in eerste instantie veel eenvoudiger, maar al snel komen de moeilijke vragen aan bod zoals data eigenaarschap, informatie-koppelingen en beveiliging.

Dit boek beschrijft wat cloud services zijn. Daarbij wordt ingegaan op de risico's die organisaties lopen bij het aangaan van contracten en SLA's.

Op basis van een lange lijst van risico's en tegenmaatregelen geeft dit boek tevens aanbevelingen voor de opzet en inhoud van de diverse service level management documenten voor cloud services. Dit boek definieert eerst het begrip 'cloud' en beschrijft daarna diverse aspecten zoals cloud patronen en de rol van een cloud broker. De kern van het boek betreft het bespreken van de contractaspecten, service documenten, service designs, risico's, SLA's en cloud governance. Om de lezer gelijk aan de slag te kunnen laten gaan met cloud SLA's zijn in het boek tevens checklists opgenomen van de volgende documenten: Underpinning Contract (UC), Service Level Agreement (SLA), Dossier Financiële Afspraken (DFA), Dossier Afspraken en Procedures (DAP), External SpecSheets (ESS) en Internal Specsheets (ISS).

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2014  
 ISBN (NL) : 978 90 7150 1739  
 ISBN (UK) : 978 94 92618 009



### SLA Templates

*A complete set of SLA templates*

The most important thing in providing a service is that the customer is satisfied with the delivered performance. With this satisfaction, the supplier gets re-purchasing's, promotions in the market and is the continuity of the company ensured. Perhaps the most important aspect of this customer satisfaction for a supplier is that the employees in question get a drive to further develop their own knowledge and skills to satisfy even more customers. This book describes the templates for Service Level Agreements in order to agree with the customer on the required service levels. This book gives both a template and an explanation for this template for all common service level management documents.

The following templates are included in this book:

- Service Level Agreement (SLA)
- Underpinning Contract (UC)
- Operational Level Agreement (OLA)
- Document Agreement and Procedures (DAP)
- Document Financial Agreements (DFA)
- Service Catalogue
- External Spec Sheet (ESS)
- Internal Spec Sheet (ISS)
- Service Quality Plan (SQP)
- Service Improvement Program (SQP)

Author : Bart de Best  
 Publisher : Leonon Media, 2017  
 ISBN (UK) : 978 94 92618 030  
 ISBN (Pocket Guide) : 978 94 92618 320



### ICT Prestatie-indicatoren

*De beheerorganisatie meetbaar gemaakt.*

De laatste jaren is het maken van concrete afspraken over de ICT-serviceverlening steeds belangrijker geworden. Belangrijke oorzaken hiervoor zijn onder meer de stringentere wet- en regelgeving, de hogere eisen die gesteld worden vanuit regievoering over uitbestede services en de toegenomen complexiteit van informatiesystemen. Om op de gewenste servicenormen te kunnen sturen, is het belangrijk om een Performance Measurement System (PMS) te ontwikkelen. Daarmee kunnen niet alleen de te leveren ICT-services worden gemeten, maar tevens de benodigde ICT-organisatie om de ICT-services te verlenen.

Het meten van prestaties is alleen zinvol als bekend is wat de doelen zijn van de opdrachtgever. Daarom start dit boek met het beschrijven van de bestuurlijke behoefte van een organisatie en de wijze waarop deze vertaald kunnen worden naar een doeltreffend PMS. Het PMS is hierbij samengesteld uit een meetinstrument voor de vakgebieden service management, project management en human resource management. Voor elk van deze gebieden zijn tevens tal van prestatie-indicatoren benoemd. Hiermee vormt dit boek een onmisbaar instrument voor zowel ICT-managers, kwaliteitsmanagers, auditors, service managers, project managers, programma managers, proces managers, als human resource managers.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2011  
 ISBN (NL) : 978 90 71501 470



### Quality Control & Assurance

*Kwaliteit op maat.*

De business stelt steeds hogere eisen aan de ICT-services die ICT-organisaties leveren. Niet alleen nemen de eisen van de overheid toe in de vorm van wet- en regelgeving, ook de dynamiek van de markt wordt hoger en de levenscyclus van business producten korter. De reactie van veel ICT-organisaties hierop is het hanteren van kwaliteitsmodellen zoals COBIT, ITIL, TOGAF en dergelijke.

Helaas verzandt het toepassen van de best practices van deze modellen vaak omdat het model als doel wordt verklaard, hierdoor ontstaat veel overhead. Nut en noodzaak worden niet onderscheiden.

In het beste geval is de borging van kwaliteit een golfbeweging met pieken en dalen waarop maar weinig grip op te krijgen is. Dit boek bespreekt op welke wijze de keuze voor kwaliteit concreet en kwantitatief gemaakt kan worden alsmede hoe de kwaliteit in de ICT-organisatie verankerd kan worden. De voorgestelde aanpak omvat zowel Quality Control (opzet en bestaan) als Quality Assurance (werking) voor ICT-processen. Hierbij worden de eisen die aan de ICT-organisatie worden gesteld vertaald naar procesrequirements (opzet) en worden deze binnen ICT-processen geborgd (bestaan). Periodiek worden deze gemeten (werking). Door requirements te classificeren naar tijd, geld, risicobeheersing en volwassenheid kan het management een bewuste keuze maken voor de toepassing van requirements. Hierdoor wordt kwaliteit meetbaar en blijft de overhead beperkt. Dit boek is een onmisbaar instrument voor kwaliteitsmanagers, auditors, lijnmanagers en proces managers.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2012  
 ISBN (NL) : 978 90 71501 531



### Acceptatiecriteria

*Naar een effectieve en efficiënte acceptatie van producten en services in de informatietechnologie.*

Acceptatiecriteria zijn een meetinstrument voor zowel gebruikers als beheerders om te bepalen of nieuwe of gewijzigde informatiesystemen voldoen aan de afgesproken requirements ten aanzien van functionaliteit, kwaliteit en beheerbaarheid. Er komt heel wat bij kijken om acceptatiecriteria te verankeren in beheerprocessen en systeemontwikkelingsprojecten. Het opstellen en het hanteren van acceptatiecriteria voor ICT-producten en ICT-services geschiedt bij veel organisaties met wisselend succes. Vaak worden acceptatiecriteria wel opgesteld, maar niet effectief gebruikt en verworpen ze tot een noodzakelijk kwaad zonder kwaliteitsborgen de werking.

Dit boek geeft een analyse van de oorzaken van dit falen van de kwaliteitsbewaking. Als remedie worden drie stappenplannen geboden voor het afleiden, toepassen en invoeren van acceptatiecriteria. De doelgroep van dit boek omvat alle partijen die betrokken zijn bij de acceptatie van ICT-producten en ICT-services: de klanten, de leveranciers en de beheerders. Ook is er nog een doelgroep die niet accepteert, maar vaststelt of correct is geaccepteerd; hiertoe behoren kwaliteitsmanagers en auditors die het boek als normenkader kunnen gebruiken. In dit boek is een aantal casussen opgenomen die diverse manieren laten zien voor het effectief en efficiënt omgaan met acceptatiecriteria.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2014  
 ISBN (NL) : 978 90 71501 784



### Beheren onder Architectuur

*Het richting geven aan de inrichting van beheerorganisaties.*

Veel organisaties zijn al jaren bezig met het vormgeven van de beheerorganisatie door vanaf de werkvloer te kijken wat er fout gaat en op basis daarvan verbetervoorstellen te formuleren. Hierbij wordt meestal gebruik gemaakt van beheermodellen, zoals ITIL, ASL en BiSL, omdat deze veel best practices bevatten. Deze bottom-up benadering werkt een lange tijd goed. De afstemming van de beheerorganisatie-inrichting op de behoefte van de business is daarmee echter nog geen feit. Het wezenlijke verschil met een top-down benadering is dat er eerst een kader gesteld wordt dat richting geeft aan de inrichting van de beheerorganisatie.

Dit kader bestaat uit beleidsuitgangspunten, architectuurprincipes en -modellen. Deze richtinggevendheid is ook van toe passing op de projectorganisatie waarin de producten en services worden vormgegeven die beheerd moeten gaan worden. Het eerste deel van dit boek positioneert dit gedachtegoed binnen de wereld van de informatievoorzieningsarchitectuur. Het tweede deel beschrijft een stappenplan om invulling te geven aan dit gedachtegoed aan de hand van vele best practices en checklists. Het derde deel beschrijft hoe beheren onder architectuur in de organisatie kan worden ingebed. Tot slot geeft het vierde deel een negental casussen van organisaties die het aangereikte stappenplan al hebben toegepast.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2017  
 ISBN (NL) : 978 90 71501 913



### Agile Service Management met Scrum

*Op weg naar een gezonde balans tussen de dynamiek van het ontwikkelen en de stabiliteit van het beheren van de informatievoorziening.*

Het toepassen van Agile software development neemt een grote vlucht. De termen Scrum en Kanban zijn al ingeburgerd bij menig organisatie. Agile software development stelt andere eisen aan de invulling van beheer van programmatuur. Veel organisaties zijn dan ook bezig om zich over deze nieuwe uitdaging te buigen. Vooral de interactie tussen het Scrum-ontwikkelpocess en het beheren van de programmatuur die het Scrum-ontwikkelpocess heeft opgeleverd is hierbij een belangrijk aspectgebied. Dit boek bespreekt juist deze interactie.

Voorbeelden van onderwerpen die hierbij ter sprake komen zijn het service portfolio, SLA's en de afhandeling van incidenten en wijzigingsverzoeken. Dit boek definieert eerst de risicogebieden bij het invoeren van Scrum en Kanban. Daarna worden de diverse Agile begrippen en concepten besproken. De invulling van Agile service management is zowel op organisatieniveau als op procesniveau beschreven. Hierbij zijn per beheerproces de relevante risico's benoemd. Tevens is aangegeven hoe hier binnen de context van Scrum invulling aan gegeven kan worden.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2014 (NL), 2018 (UK)  
 ISBN (NL) : 978 90 7150 1807  
 ISBN (UK) : 978 94 92618 085



### Agile Service Management met Scrum in de Praktijk

*Op weg naar een gezonde balans tussen de dynamiek van het ontwikkelen en de stabiliteit van het beheren van de informatievoorziening.*

Veel bedrijven zijn bezig om Agile softwareontwikkeling toe te gaan passen in de vorm van Scrum of Kanban of hebben het nieuwe ontwikkelproces al in gebruik genomen. Vroeg of laat komt dan de vraag hoe dit ontwikkelproces zich verhoudt tot de beheerprocessen. In het boek 'Agile Service Management met Scrum' is al naar deze interface gekeken en zijn een aantal risico's per beheerproces onderkend. Tevens zijn tegenmaatregelen gedefinieerd die genomen kunnen worden.

In een onderzoek bij tien organisaties zijn deze risico's voorgelegd en is gevraagd hoe zij met deze risico's zijn omgegaan. Tevens is onderzocht welke Agile aspecten worden toegepast en in het bijzonder die van Scrum of Kanban. Tot slot is door elke organisatie een volwassenheidsassessment uitgevoerd voor zowel het Agile ontwikkelproces als het change management proces. Dit boek is het rapport over het onderzoek naar de samenwerking van Agile software ontwikkeling en beheerprocessen in de praktijk. De doelgroep van dit boek omvat alle partijen die betrokken zijn bij de toepassing van Agile software ontwikkeling en die graag eens willen weten hoe collega's deze cruciale interface voor een succesvolle serviceverlening hebben vormgegeven. In dit boek is tevens van elke organisatie een korte beschrijving gegeven over de wijze waarop het Agile ontwikkelproces is vormgegeven.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2015 (NL), 2018 (UK)  
 ISBN (NL) : 978 90 7150 1845  
 ISBN (UK) : 978 94 92618 177





### DevOps Best Practices

*Best Practices for DevOps*

In recent years, many organisations have experienced the benefits of using Agile approaches such as Scrum and Kanban. The software is delivered faster whilst quality increases and costs decrease. The fact that many organisations that applied the Agile approach did not take into account the traditional service management techniques, in terms of information management, application management and infrastructure management, is a major disadvantage. The solution to this problem has been found in the Dev (Development) Ops (Operations) approach. Both worlds are merged into one team, thus sharing the knowledge and skills. This book is about sharing knowledge on how DevOps teams work together.

For each aspect of the DevOps process best practices are given in 30 separate articles. The covered aspects are Plan, Code, Build, Test, Release, Deploy, Operate and Monitor. Each article starts with the definition of the specifically used terms and one or more concepts. The body of each article is kept simple, short, and easy to read.

Author : Bart de Best  
 Publisher : Leonon Media, 2017 (UK), 2018 (UK)  
 ISBN (UK) : 978 94 92618 078  
 ISBN (Pocket Guide) : 978 94 92618 306



### DevOps Architectuur

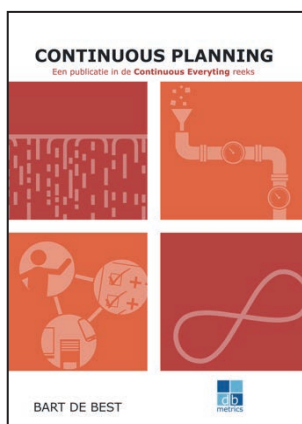
*DevOps Architectuur Best Practices*

De wereld van systeemontwikkeling is in een snel tempo aan het veranderen. Daarbij worden Development (Dev) en Operations (Ops) steeds meer geïntegreerd zodat oplossingen sneller en kwalitatief beter aan de klant kunnen worden aangeboden. De vraag is hoe binnen deze nieuwe zienswijze van DevOps plaats is voor Agile architectuur. Dit boek geeft een antwoord op deze vraag door het geven van vele voorbeelden van architectuurprincipes- en modellen die richting geven aan de inrichting en de verrichting van een DevOps organisatie. In het hele boek wordt zo veel als mogelijk per paragraaf een toelichting gegeven op basis van een denkbeeldig bedrijf Assuritas.

Dit boek bestaat uit verschillende onderdelen hetgeen het boek modulair maakt. Het hoeft dus niet van A tot Z gelezen te worden. Na de korte schets van het casusbedrijf volgt de bespreking van de DevOps organisatie vanuit een architectuurperspectief. Daarna wordt de DevOps beheervoorziening besproken. Beide verhandelingen worden aan de hand van het casusbedrijf inzichtelijk gemaakt. Na de behandeling van de integratie van de Dev- en Ops-rollen volgen twee handige analysetools om de volwassenheid van DevOps te bepalen. Het boek sluit af met een casus waarin op basis van architectuurprincipes en -modellen de keuze voor een Agile documentatie wordt gemaakt. Dit werk over DevOps architectuur is een onmisbaar hulpmiddel bij de vormgeving en uitvoering van een DevOps serviceorganisatie.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2019  
 ISBN (NL) : 978 94 92618 061  
 ISBN (UK) : 978 90 71501 579

## Continuous Everything boeken



### Continuous Planning

Een uitgave in de Continuous Everything reeks.

Continuous planning is een aanpak om grip te krijgen op veranderingen die aangebracht worden in de informatievoorziening teneinde de outcome verbetering van de bedrijfsprocessen te realiseren en daarmee de bedrijfsdoelen te behalen. De aanpak is gericht op meer niveaus waarbij voor elke niveau een Agile planningstechniek wordt aangereikt die de hoger liggende planning verfijnt. Op deze manier kan er zowel op strategisch, tactisch als operationeel niveau een planning worden gemaakt en wel op een Agile wijze die zo min mogelijk overhead en zoveel mogelijk waarde creëren. Dit boek is een publicatie in de continuous everything reeks. De content bestaat uit de bespreking van de plannings-technieken zoals de balanced scorecard, enterprise architectuur,

product vision, roadmap, epic one pager, product backlog management, release planning en sprint planning. Tevens wordt aangegeven hoe deze technieken aan elkaar zijn gerelateerd. Daarnaast geeft dit boek aan hoe continuus planning in te richten in uw organisatie op basis van het paradigma van de verandermanager en architectuurprincipes en -modellen. Met deze integrale Agile benadering van planning heeft u een krachtig gereedschap in handen om de strategie van uw organisatie planmatig op te pakken en daarmee uw bedrijfsdoelen te realiseren.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2022  
 ISBN (NL) : 978 94 92618 504  
 ISBN (UK) : 978 94 92618 726



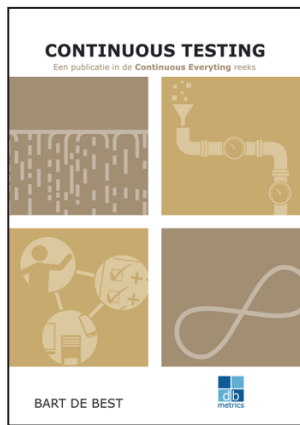
### Continuous Design

Een uitgave in de Continuous Everything reeks.

Continuous design is een aanpak die beoogt om DevOps teams vooraf kort na te laten denken over de contouren van het te realiseren informatiesysteem en tijdens het Agile project het design te laten groeien (emerging design). Hierdoor worden interface risico's voorkomen en wordt essentiële kennisoverdracht geborgd ter ondersteuning van beheer en het nakomen van wet- en regelgeving. Elementen die de continuïteit van een organisatie waarborgen. Dit boek is een publicatie in de continuous everything reeks. De content bestaat uit het continuous design pyramid model waarin de volgende design views zijn gedefinieerd: business, solution, design, requirements, test en code view.

Het continuous design omvat de gehele lifecycle van het informatiesysteem. De eerste drie views worden ingevuld op basis van moderne ontwerpstechnieken zoals de value stream mapping en use cases. De nadruk van het effectief toepassen van een continuus design ligt echter in de realisatie van het informatiesysteem en wel door het design te integreren in de Behaviour Driven Development en Test Driven Development alsmede in continuus documentation. Met deze Agile benadering van een design heeft u een krachtig gereedschap in handen om grip te krijgen op een Agile ontwikkelproject.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2022  
 ISBN (NL) : 978 94 92618 481  
 ISBN (UK) : 978 94 92618 702



### Continuous Testing

Een uitgave in de Continuous Everything reeks.

Continuous testing is een aanpak die beoogt om een fast feedback te geven in het software ontwikkelproces door de 'wat'- en 'hoe'-vragen te definiëren als testcases voordat gestart wordt met de bouw van de oplossing. Hierdoor worden de concepten requirements, testcases en acceptatiecriteria geïntegreerd in één aanpak. Het begrip 'continuus' verwijst naar het toepassen van test management in alle fasen van de deployment pipeline, dus van requirements tot en met het in productie nemen. Tevens omvat het begrip 'continuus' de aspecten People, Process en Technology. Daarmee wordt test management dus holistisch. Dit boek is een publicatie in de continuous everything reeks.

De content bestaat uit het behandelen van continuous testing aan de hand van een definitie, business case, architectuur, ontwerp en best practices. Begrippen die hierbij worden besproken zijn: het veranderparadigma, de ideal test pyramid, test meta data, Behaviour Driven Development, Test Driven Development, test policies, test technieken, test tools en de rol van unit testcases in continuous testing. Op deze wijze bent u snel weer up-to-date op het gebied van de DevOps ontwikkelingen en op het gebied van continuous testing.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2022  
 ISBN (NL) : 978 94 92618 450  
 ISBN (UK) : 978 94 92618 672



### Continuous Integration

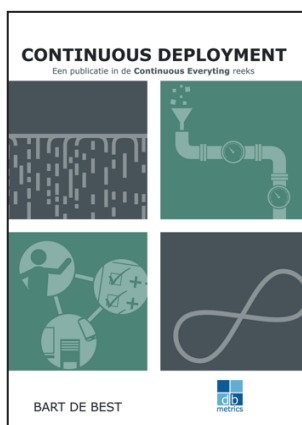
Een uitgave in de Continuous Everything reeks.

Continuous integration is een holistische Lean software ontwikkel-aanpak die beoogt om op een incrementele en iteratieve wijze continu software te produceren en in productie te nemen waarbij waste reductie hoog in het vaandel staat.

Het woord 'holistisch' verwijst naar de PPT-begrippen: People (multiple expert), Process (kennis van bedrijfs- en beheer-processen) en Technology (applicatie en infrastructuur programmeren). Door de incrementele en iteratieve werkwijze wordt fast feedback mogelijk doordat functionaliteiten eerder in productie kunnen worden genomen. Hierdoor wordt waste gereduceerd omdat gebreken eerder worden gevonden en sneller kunnen

worden hersteld. Dit boek is een publicatie in de continuous everything reeks. De content bestaat uit het behandelen van continuous integration aan de hand van een definitie, business case, architectuur, ontwerp en best practices. Begrippen die hierbij worden besproken zijn het veranderparadigma, het toepassen van continuous integration, gebruik repositories, code kwaliteit, green code, green build, refactoring, security based development en built-in failure mode. Op deze wijze bent u snel weer up-to-date op het gebied van de DevOps ontwikkelingen met betrekking tot continuous integration.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2022  
 ISBN (NL) : 978 94 92618 467  
 ISBN (UK) : 978 94 92618 689



## Continuous Deployment

Een uitgave in de Continuous Everything reeks.

Continuous deployment is een holistische Lean production aanpak die beoogt om op een incrementele en iteratieve wijze continu software te deployen en te releasen waarbij time to market en hoogwaardige kwaliteit hoog in het vaandel staan.

Het woord 'holistisch' verwijst naar de PPT-begrippen: People (multiple expert), Process (kennis van bedrijfs- en beheerprocessen) en Technology (applicatie en infrastructuur programmeren). Door de incrementele en iteratieve deployments wordt fast feedback mogelijk omdat fouten eerder in productie van de CI/CD secure pipeline worden waargenomen. Hierdoor zijn herstelacties sneller en goedkoper hetgeen leidt tot een waste reductie.

Dit boek is een publicatie in de continuous everything reeks. De content bestaat uit het behandelen van continuous deployment aan de hand van een definitie, business case, architectuur, ontwerp en best practices. Begrippen die hierbij worden besproken zijn het veranderparadigma, het toepassen van continuous deployment, een stappenplan voor het planmatig inregelen van continuous deployment en vele patterns om deployments te laten plaatsvinden. Op deze wijze bent u snel weer up-to-date op het gebied van de DevOps ontwikkelingen op het gebied van continuous deployment.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2022  
 ISBN (NL) : 978 94 92618 511  
 ISBN (UK) : 978 94 92618 733



## Continuous Monitoring

Een uitgave in de Continuous Everything reeks.

Continuous monitoring is een aanpak om grip te krijgen op zowel core value streams (business processen) als enable value streams die deze core value streams ondersteunen. Continuous monitoring onderscheidt zich van de klassieke monitoring door de focus op de outcome verbetering en de holistisch scope waarmee value streams worden gemeten te weten de gehele CI/CD secure pipeline voor alle drie de perspectieven van PPT: People, Process en Technology.

De aanpak omvat People, Process en Technology, hetgeen het mogelijk maakt om de bottlenecks in uw value streams in kaart te brengen en te elimineren of te mitigeren.

Dit boek is een publicatie in de continuous everything reeks. De content bestaat uit de bespreking van de monitorfuncties die zijn gedefinieerd in het continuous monitoring lagenmodel. Dit lagenmodel classificeert de op de markt beschikbare monitortools. Elk monitor archetype wordt in dit boek gedefinieerd qua definitie, doelstelling, meetattributen, requirements, voorbeelden en best practices. Tevens geeft dit boek aan hoe continuous monitoring in te richten in uw organisatie op basis van het paradigma van de verandermanager en architectuurprincipes en -modellen. Met deze integrale Agile benadering van monitoring heeft u een krachtig gereedschap in handen om de controls in te regelen voor de besturing van uw value streams.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2022  
 ISBN (NL) : 978 94 92618 498  
 ISBN (UK) : 978 94 92618 719



### Continuous Learning

Een uitgave in de Continuous Everything reeks.

Continuous learning is een aanpak om grip te krijgen op de competenties die nodig zijn om de strategie van uw organisatie te realiseren.

Continuous learning biedt Human Resource Management hiertoe een aanpak die stap voor stap de organisatiebehoefte competenties verkent en deze behoeften omzet in competentieprofielen. Een competentieprofiel is hierbij gedefinieerd als de set van kennis, kunde en gedrag op een bepaald Bloom level die een bepaald resultaat oplevert. Competentieprofielen worden vervolgens samengevoegd in rollen die op hun beurt functies vormen.

Op deze wijze wordt een Agile functiehuis verkregen. Dit boek is een publicatie in de continuous everything reeks. De content bestaat uit de bespreking van het continuous learning model dat u helpt om stap voor stap een value chain strategie naar een persoonlijke roadmaps voor medewerkers te vertalen. Tevens geeft dit boek aan hoe continuous learning in te richten in uw organisatie op basis van het paradigma van de verandermanager en architectuurprincipes en -modellen. Met deze Agile benadering van HRM heeft u een krachtig gereedschap in handen om de competenties op het gewenste niveau van uw organisatie te krijgen.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2022  
 ISBN (NL) : 978 94 92618 528  
 ISBN (UK) : 978 94 92618 740



### Continuous Assessment

Een uitgave in de Continuous Everything reeks.

Continuous assessment is een aanpak die beoogt om DevOps teams zich op een continue wijze zich te laten ontwikkelen qua kennis en kunde op het gebied van business, development, operations en security.

Dit boek geeft een hulpmiddel om de DevOps teams bewust te maken waar zij staan qua ontwikkeling en welke eerstvolgende stappen zij kunnen zetten om zich te ontwikkelen. Dit boek is een publicatie in de continuous everything reeks. De content bestaat uit de business case voor continuous assessment, de architectuur van de twee assessment modellen en de assessment vragenlijsten.

Het DevOps Cube model is gebaseerd op het idee dat DevOps vanuit zes verschillende perspectieven van een kubus kunnen worden bekeken te weten: 'Flow', 'Feedback', 'continuous learning', 'Governance', 'Pipeline' en 'QA'. Het DevOps CE model is gebaseerd op de continuous everything perspectieven te weten: 'continuous integration', 'continuous deployment', 'continuous testing', 'continuous monitoring', 'continuous documentation' en 'continuous learning'. Dit boek is een uitstekende spiegel voor ieder DevOps team dat snel een compleet beeld wil vormen van op te pakken DevOps best practices.

Auteur : Bart de Best  
 Uitgever : Leonon Media, 2022  
 ISBN (NL) : 978 94 92618 474  
 ISBN (UK) : 978 94 92618 696



## Continuous Auditing

Een uitgave in de Continuous Everything reeks.

Continuous auditing is een aanpak die beoogt om DevOps teams in staat te stellen kort cyclisch aan te tonen in control te zijn bij het in hoog tempo realiseren, in productie nemen en beheren van de nieuwe of aangepaste producten en services.

Hierdoor worden compliancy risico's voorkomen door al vanuit de requirements en het daarop gebaseerde design na te denken over welke risico's te mitigeren of te elimineren. Dit boek is een publicatie in de continuous everything reeks.

De content bestaat uit de bespreking van de continuous auditing pyramid model die de zes stappen beschrijft om continuous auditing invulling te geven te weten: scope bepalen, doelen bepalen, risico's identificeren, controls realiseren, monitorvoorziening inrichten en effectiviteit controls aantonen. Het continuous auditing concept omvat hiermee de gehele lifecycle van de risicobeheersing. Hierdoor zijn de risico's continu in control. Met deze Agile benadering van auditing heeft u een krachtig gereedschap in handen om grip te krijgen op de compliancy van uw Agile systeemontwikkeling -en beheer.

Author : Bart de Best  
 Publisher : Leonon Media, 2022  
 ISBN (NL) : 978 94 92618 542  
 ISBN (UK) : 978 94 92618 818



## Continuous Security

Een uitgave in de Continuous Everything reeks.

Continuous security is een aanpak die beoogt om organisatie continue in control te laten zijn vanuit drie perspectieven:

- Het business perspectief: Business value streams zijn in control voor de onderkende risico's door de effectiviteit van de ingezette controls continue te toetsen en evidence vast te leggen.
- Het development perspectief: Development value streams zijn in control door de non functionele requirements voor information security integraal mee te nemen in de ontwikkeling.
- Het operations perspectief: Operations value streams zijn in control voor de voortbrenging van de nieuwe en aangepaste ICT services door een adequate inrichting van de CI/CD secure

pipeline waarin controls automatisch de non functionele requirements toetsen. Dit boek is een publicatie in de Continuous Everything reeks. De content bestaat uit de bespreking van de toepassing van ISO 27001 aan de hand van drie sets van security practices te weten Governance, Risk en Quality. De practices zijn voorzien van een definitie en doelstelling. Daarnaast worden voorbeelden en best practices gegeven.

Het continuous security concept is ontworpen om gebruikt te worden in Agile Scrum (development) en DevOps (development & operations) omgevingen. Daartoe sluit het naadloos aan op gangbare Agile beheermodellen. Met deze Agile benadering van information security heeft u een krachtig gereedschap in handen om grip te krijgen op de compliancy van uw Agile systeemontwikkeling -en beheer.

Author : Bart de Best  
 Publisher : Leonon Media, 2022  
 ISBN (NL) : 978 94 91480 171  
 ISBN (UK) : 978 94 91480 188



### Continuous Development

Een uitgave in de Continuous Everything reeks.

Continuous Everything is de verzamelnaam van alle Continuous ontwikkelingen die er momenteel gaande zijn in de DevOps wereld. Door deze onder één noemer te laten vallen kan er structuur worden aangebracht aan de individuele ontwikkelingen en kunnen op basis van patterns best practices worden gedefinieerd.

Het begrip 'Continuous' omvat de termen: outcome driven development, incrementeel & iteratief werken, waste reductie door een Lean aanpak, holistisch werken door people, process, partner & technology in de scope mee te nemen en continue aandacht te geven aan een op te leveren product of service in de hele levenscyclus vanuit een end-to-end benadering.

Dit boek is een bundeling van vier Continuous Everything boeken te weten: Continuous Planning, Continuous Design, Continuous Testing en Continuous Integration. Voor elk Continuous Everything aspectgebied wordt aangegeven hoe deze in te richten in uw organisatie op basis van het paradigma van de verandermanager en architectuurprincipes en -modellen. Tevens worden per aspectgebied de best practices besproken. Met dit boek in de hand heeft u een krachtig gereedschap om u verder te bekwamen op het gebied van DevOps.

Author : Bart de Best  
 Publisher : Leonon Media, 2022  
 ISBN (NL) : 978 94 92618 641  
 ISBN (UK) : 978 94 92618 764



### Continuous Operations

Een uitgave in de Continuous Everything reeks.

Continuous Everything is de verzamelnaam van alle Continuous ontwikkelingen die er momenteel gaande zijn in de DevOps wereld. Door deze onder één noemer te laten vallen kan er structuur worden aangebracht aan de individuele ontwikkelingen en kunnen op basis van patterns best practices worden gedefinieerd.

Het begrip 'Continuous' omvat de termen: outcome driven development, incrementeel & iteratief werken, waste reductie door een Lean aanpak, holistisch werken door people, process, partner & technology in de scope mee te nemen en continue aandacht te geven aan een op te leveren product of service in de hele levenscyclus vanuit een end-to-end benadering.

Dit boek is een bundeling van vier Continuous Everything boeken te weten: Continuous Deployment, Continuous Monitoring, Continuous Learning en Continuous Assessment. Voor elk Continuous Everything aspectgebied wordt aangegeven hoe deze in te richten in uw organisatie op basis van het paradigma van de verandermanager en architectuurprincipes en -modellen. Tevens worden per aspectgebied de best practices besproken. Met dit boek in de hand heeft u een krachtig gereedschap om u verder te bekwamen op het gebied van DevOps.

Author : Bart de Best  
 Publisher : Leonon Media, 2022  
 ISBN (NL) : 978 94 92618 658  
 ISBN (UK) : 978 94 92618 771



### Continuous Control

Een uitgave in de Continuous Everything reeks.

Continuous Everything is de verzamelnaam van alle Continuous ontwikkelingen die er momenteel gaande zijn in de DevOps wereld. Door deze onder één noemer te laten vallen kan er structuur worden aangebracht aan de individuele ontwikkelingen en kunnen op basis van patterns best practices worden gedefinieerd.

Het begrip 'Continuous' omvat de termen: outcome driven development, incrementeel & iteratief werken, waste reductie door een Lean aanpak, holistisch werken door people, process, partner & technology in de scope mee te nemen en continue aandacht geven aan een op te leveren product of service in de hele levenscyclus vanuit een end-to-end benadering.

Dit boek is een bundeling van drie Continuous Everything boeken te weten: Continuous Assessment, Continuous Security en Continuous Audit. Voor elk Continuous Everything aspectgebied wordt aangegeven hoe deze in te richten in uw organisatie op basis van het paradigma van de verandermanager en architectuurprincipes en -modellen. Tevens worden per aspectgebied de best practices besproken. Met dit boek in de hand heeft u een krachtig gereedschap om u verder te bekwamen op het gebied van DevOps.

Author : Bart de Best  
 Publisher : Leonon Media, 2022  
 ISBN (NL) : 978 94 91480 195  
 ISBN (UK) : 978 94 91480 201



### Continuous Everything

Een uitgave in de Continuous Everything reeks.

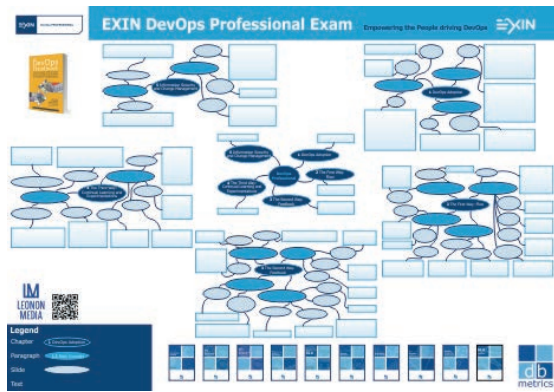
Continuous Everything is de verzamelnaam van alle Continuous ontwikkelingen die er momenteel gaande zijn in de DevOps wereld. Door deze onder één noemer te laten vallen kan er structuur worden aangebracht aan de individuele ontwikkelingen en kunnen op basis van patterns best practices worden gedefinieerd.

Het begrip 'Continuous' omvat de termen: outcome driven development, incrementeel & iteratief werken, waste reductie door een Lean aanpak, holistisch werken door people, process, partner & technology in de scope mee te nemen en continue aandacht te geven aan een op te leveren product of service in de hele levenscyclus vanuit een end-to-end benadering.

Dit boek is een bundeling van acht Continuous Everything boeken te weten: Continuous Planning, Continuous Design, Continuous Testing, Continuous Integration, Continuous Deployment, Continuous Monitoring, Continuous Learning en Continuous Assessment. Voor elk Continuous Everything aspectgebied wordt aangegeven hoe deze in te richten in uw organisatie op basis van het paradigma van de verandermanager en architectuurprincipes en -modellen. Tevens worden per aspectgebied de best practices besproken. Met dit boek in de hand heeft u een krachtig gereedschap om u verder te bekwamen op het gebied van DevOps.

Author : Bart de Best  
 Publisher : Leonon Media, 2022  
 ISBN (NL) : 978 94 92618 597  
 ISBN (UK) : 978 94 92618 665





### DevOps Poster

#### *DevOps Professional Exam Poster*

This poster lists all the DevOps terms that a student must learn in order to pass the exam of DevOps Professional of Exin. This poster can be ordered at [info@leonon.nl](mailto:info@leonon.nl).

The subjects on the poster are based on the basic training material of Exin. Since there are many terms to be learned, this poster will help to learn them by reviewing them all at once daily.

Author : Bart de Best  
 Publisher : Leonon Media, 2018  
 Ordering : [info@leonon.nl](mailto:info@leonon.nl)

# CONTINUOUS DESIGN

Een publicatie in de  
**Continuous Everything**  
reeks

Bart de Best



**Continuous Design is een aanpak die beoogt om DevOps teams vooraf kort na te laten denken over de contouren van het te realiseren informatiesysteem en tijdens het Agile project het design te laten groeien (emerging design).**

**Hierdoor worden interface risico's voorkomen en wordt essentiële kennisoverdracht geborgd ter ondersteuning van beheer en het nakomen van wet- en regelgeving. Elementen die de continuïteit van een organisatie waarborgen.**

Dit boek is een publicatie in de Continuous Everything reeks. De content bestaat uit het Continuous Design Pyramid model waarin de volgende design views zijn gedefinieerd: business, solution, design, requirements, test en code view.

Het Continuous Design omvat de gehele lifecycle van het informatiesysteem. De eerste drie views worden ingevuld op basis van moderne ontwerptechnieken zoals de value stream mapping en use cases. De nadruk van het effectief toepassen van een Continuous Design ligt echter in de realisatie van het informatiesysteem en wel door het design te integreren in de Behaviour Driven Development en Test Driven Development alsmede in Continuous Documentation.

Met deze Agile benadering van een design heeft u een krachtig gereedschap in handen om grip te krijgen op een Agile ontwikkelproject.

ISBN 978-94-92618-48-1

