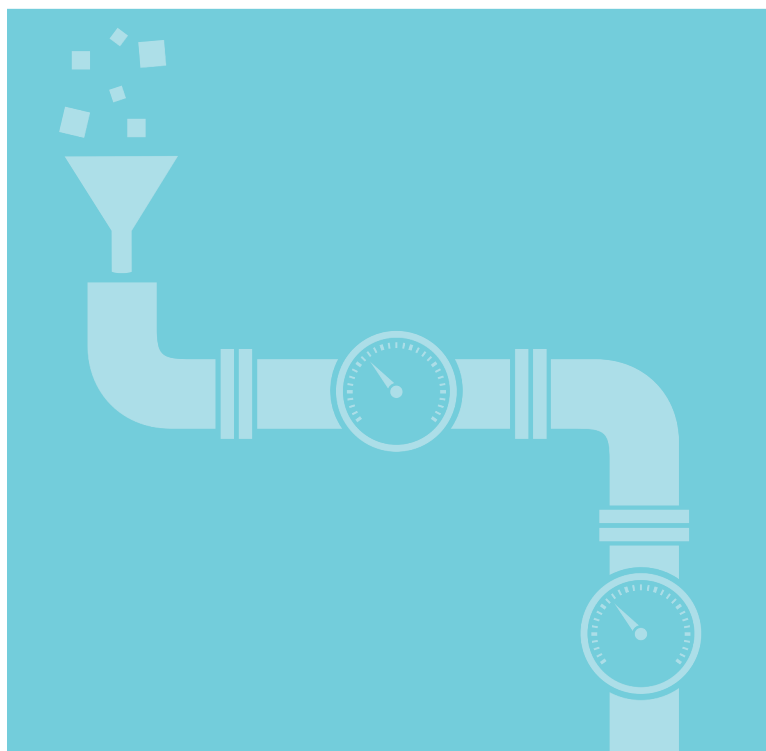


CONTINUOUS OPERATIONS

Het volledige ABC van DevOps Operations



BART DE BEST



DevOps Continuous Development

Het volledige ABC van DevOps Development

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

© 2023 Leonon Media

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

Titel : DevOps Continuous Development
Sub titel : Het volledige ABC van DevOps Development
Datum : 18 januari 2023
Auteur : Bart de Best
Uitgever : Leonon Media
ISBN13 : 978 94 92618 641
Druk : Eerste druk, zevende editie, 18 januari 2023

©2023, 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

1	INLEIDING	1
1.1	DOELSTELLING.....	1
1.2	DOELGROEP.....	1
1.3	ACHTERGROND	1
1.4	STRUCTUUR	2
1.5	BIJLAGEN	3
1.6	LEESWIJZER.....	3

DEEL I. CONTINUOUS PLANNING

1	INLEIDING CONTINUOUS PLANNING	7
1.1	DOEL.....	7
1.2	POSITIONERING	7
1.3	STRUCTUUR	7
2	BASISCONCEPTEN EN BASISBEGRIPPEN	9
2.1	BASISCONCEPTEN	9
2.2	BASISBEGRIPPEN.....	11
3	CONTINUOUS PLANNING DEFINITIE	17
3.1	ACHTERGROND	17
3.2	DEFINITIE	17
3.3	TOEPASSING	17
4	CONTINUOUS PLANNING VERANKERING	21
4.1	HET VERANDERPARADIGMA	21
4.2	BEELDVORMING	22
4.3	MACHTSVERHOUDING	23
4.4	ORGANISATIEVORMGEVING	26
4.5	RESOURCES	27
5	CONTINUOUS PLANNING ARCHITECTUUR	29
5.1	ARCHITECTUURPRINCIPES.....	29
5.2	ARCHITECTUURMODELLEN.....	32
6	CONTINUOUS PLANNING ONTWERP	35
6.1	CONTINUOUS PLANNING VALUE STREAM.....	35
6.2	CONTINUOUS PLANNING USE CASE DIAGRAM	35
6.3	CONTINUOUS PLANNING USE CASE	36
7	CONTINUOUS PLANNING MODEL	41
7.1	INLEIDING.....	41
7.2	BALANCED SCORECARD	41
7.3	ENTERPRISE ARCHITECTUUR	46
7.4	PRODUCT VISION.....	48
7.5	ROADMAP	53
7.6	PRODUCT BACKLOG	55
7.7	RELEASE PLAN	55
7.8	SPRINT PLANNING	56

DEEL II. CONTINUOUS DESIGN

1	INLEIDING CONTINUOUS DESIGN	61
1.1	DOEL	61
1.2	POSITIONERING.....	61
1.3	STRUCTUUR.....	62
2	BASISCONCEPTEN EN BASISBEGRIPPEN	65
2.1	BASISCONCEPTEN	65
2.2	BASISBEGRIPPEN	66
3	CONTINUOUS DESIGN DEFINITIE	69
3.1	ACHTERGROND	69
3.2	DEFINITIE	69
3.3	TOEPASSING	69
4	CONTINUOUS DESIGN VERANKERING	71
4.1	HET VERANDERPARADIGMA	71
4.2	BEELDVORMING	72
4.3	MACHTSVERHOUDING.....	73
4.4	ORGANISATIEVORMGEVING.....	76
4.5	RESOURCES	78
4.6	WATERVAL VERSUS CONTINUOUS DESIGN	79
5	CONTINUOUS DESIGN ARCHITECTUUR	83
5.1	ARCHITECTUURPRINCIPES	83
5.2	ARCHITECTUURMODELLEN	85
6	CONTINUOUS DESIGN ONTWERP	87
6.1	CONTINUOUS DESIGN VALUE STREAM	87
6.2	CONTINUOUS DESIGN USE CASE DIAGRAM.....	87
6.3	CONTINUOUS DESIGN USE CASE	88
7	BUSINESS VIEW	93
7.1	INLEIDING	93
7.2	SYSTEM CONTEXT DIAGRAM	93
7.3	VALUE STREAM CANVAS.....	96
8	SOLUTION VIEW	101
8.1	INLEIDING	101
8.2	USE CASE DIAGRAM	101
8.3	SYSTEM BUILDING BLOCKS.....	105
8.4	VALUE STREAM MAPPING	116
9	DESIGN VIEW	119
9.1	INLEIDING	119
9.2	USE CASE.....	120
10	REQUIREMENT VIEW	127
10.1	INLEIDING	127
10.2	BEHAVIOR DRIVEN DEVELOPMENT	128
11	TEST VIEW	131
11.1	INLEIDING	131
11.2	TEST DRIVEN DEVELOPMENT.....	132

12	CODE VIEW	135
12.1	INLEIDING.....	135
12.2	CONTINUOUS DOCUMENTATION DEVELOPMENT.....	136
13	CONTINUOUS DESIGN BIJ ASSURITAS.....	143
13.1	ASSURITAS.....	143
13.2	CONTINUOUS DESIGN.....	145

DEEL III. CONTINUOUS TESTING

1	INLEIDING CONTINUOUS TESTING.....	153
1.1	DOEL.....	153
1.2	POSITIONERING.....	153
1.3	STRUCTUUR.....	153
2	BASISCONCEPTEN EN BASISCONCEPTEN.....	155
2.1	BASISCONCEPTEN.....	155
2.2	BASISBEGRIPPEN.....	155
3	CONTINUOUS TESTING DEFINITIE.....	159
3.1	ACHTERGROND.....	159
3.2	DEFINITIE.....	159
3.3	TOEPASSING.....	160
4	CONTINUOUS TESTING VERANKERING.....	161
4.1	HET VERANDERPARADIGMA.....	161
4.2	BEELDVORMING.....	162
4.3	MACHTSVERHOUDING.....	164
4.4	ORGANISATIEVORMGEVING.....	167
4.5	RESOURCES.....	169
5	CONTINUOUS TESTING ARCHITECTUUR.....	173
5.1	ARCHITECTUUR PRINCIPES.....	173
5.2	FAST EN LATE FEEDBACK.....	176
5.3	TESTSOORT-MATRIX.....	177
5.4	TESTTECHNIEK-MATRIX.....	178
5.5	TESTOBJECT-MATRIX.....	180
5.6	TESTTOOL-MATRIX.....	181
6	CONTINUOUS TESTING ONTWERP.....	183
6.1	CONTINUOUS TESTING VALUE STREAM.....	183
6.2	CONTINUOUS TESTING USE CASE DIAGRAM.....	184
6.3	CONTINUOUS TESTING USE CASE.....	185
7	CONTINUOUS TESTING BEST PRACTICES.....	191
7.1	BEHAVIOR DRIVEN DEVELOPMENT.....	191
7.2	TEST DRIVEN DEVELOPMENT.....	193
7.3	UNITTEST POLICIES.....	194
7.4	GENERIEKE TESTSTRATEGIE.....	195
7.5	OVERIGE.....	196

DEEL IV. CONTINUOUS INTEGRATION

1	INLEIDING CONTINUOUS INTEGRATION	201
1.1	DOEL	201
1.2	POSITIONERING.....	201
1.3	STRUCTUUR.....	201
2	BASISCONCEPTEN EN BASISBEGRIPPEN	203
2.1	BASISCONCEPTEN	203
2.2	BASISBEGRIPPEN	205
3	CONTINUOUS INTEGRATION DEFINITIE.....	209
3.1	ACHTERGROND	209
3.2	DEFINITIE	210
3.3	TOEPASSING	210
4	CONTINUOUS INTEGRATION VERANKERING	213
4.1	HET VERANDERPARADIGMA	213
4.2	BEELDVORMING	214
4.3	MACHTSVERHOUDING.....	216
4.4	ORGANISATIEVORMGEVING.....	219
4.5	RESOURCES	222
5	CONTINUOUS INTEGRATION ARCHITECTUUR	225
5.1	ARCHITECTUURPRINCIPES	225
5.2	VERSIEBEHEER	227
5.3	FAST EN LATE FEEDBACK	235
6	CONTINUOUS INTEGRATION ONTWERP	237
6.1	CONTINUOUS INTEGRATION VALUE STREAM	237
6.2	CONTINUOUS INTEGRATION USE CASE DIAGRAM.....	237
6.3	CONTINUOUS INTEGRATION USE CASE	238
7	CONTINUOUS INTEGRATION BEST PRACTICES	243
7.1	CONTINUOUS INTEGRATION STAPPENPLAN	243
7.2	SAMENWERKING – BRANCHING & MERGING	244
7.3	SAMENWERKING – GREEN BUILD	248
7.4	SAMENWERKING – KAIZEN.....	248
7.5	CODE KWALITEIT	249
7.6	NON FUNCTIONAL REQUIREMENTS	250
7.7	OVERIGE.....	251
	BIJLAGE A, LITERATUURLIJST	257
	BIJLAGE B, BEGRIPPENLIJST	261
	BIJLAGE C, AFKORTINGEN	277
	BIJLAGE D, WEBSITES.....	281
	BIJLAGE E, INDEX	283

Figuren

FIGUUR 1-1, DEVOPS LEMNISCAAT.	1
-------------------------------------	---

DEEL I. CONTINUOUS PLANNING

FIGUUR 2-1, ROADMAP TO VALUE MODEL.	9
FIGUUR 2-2, CONTINUOUS PLANNING MODEL.	10
FIGUUR 2-3, PLANNING & DESIGN MODEL.	10
FIGUUR 2-4, BALANCED SCORE CARD [KAPLAN 2004].	11
FIGUUR 2-5, ENTERPRISE ARCHITECTURE.	12
FIGUUR 2-6, ROADMAP.	13
FIGUUR 2-7, VALUE CHAIN OF PORTER, BRON: [BOEK MICHAEL PORTER].	14
FIGUUR 2-8, RECURSIEVE VALUE CHAIN OF PORTER, BRON: [BOEK MICHAEL PORTER].	15
FIGUUR 2-9, RECURSIEVE VALUE CHAIN OF PORTER, BRON: [BOEK MICHAEL PORTER].	15
FIGUUR 2-10, SoR EN SoE, BRON HSO THE RESULT COMPANY.	16
FIGUUR 4-1, VERANDERPARADIGMA.	21
FIGUUR 4-2, VERANDERPARADIGMA - BEELDVORMING.	22
FIGUUR 4-3, VERANDERPARADIGMA - MACHTSVERHOUDING.	24
FIGUUR 4-4, VERANDERPARADIGMA - ORGANISATIE.	26
FIGUUR 4-5, VERANDERPARADIGMA - RESOURCES.	28
FIGUUR 5-1, ROADMAP TO VALUE, BRON: [LAYTON 2017].	33
FIGUUR 5-2, CONTINUOUS PLANNING MODEL.	33
FIGUUR 5-3, PLANNING & DESIGN MODEL.	34
FIGUUR 6-1, CONTINUOUS PLANNING VALUE STREAM.	35
FIGUUR 6-2, USE CASE DIAGRAM VOOR INRICHTING VAN CONTINUOUS PLANNING.	36
FIGUUR 7-1, CONTINUOUS PLANNING MODEL.	41
FIGUUR 7-2, BALANCED SCORECARD.	42
FIGUUR 7-3, GECASCADEERDE BALANCED SCORECARD.	43
FIGUUR 7-4, VOORBEELD VAN EEN INGEVULDE BALANCED SCORECARD.	44
FIGUUR 7-5, OPERATING MODEL.	45
FIGUUR 7-6, EEN ENTERPRISE ARCHITECTUURPLAAT.	46
FIGUUR 7-7, TEMPLATE SYSTEM BUILDING BLOCKS – ALGEMEEN.	47
FIGUUR 7-8, VOORBEELD VAN INFORMATION SYSTEM BUILDING BLOCKS VAN EEN KOFFIESERVICE.	48
FIGUUR 7-9, RELEASEPLAN.	56
FIGUUR 7-10, DEFINITION OF DONE.	58

DEEL II. CONTINUOUS DESIGN

FIGUUR 1-1, SoR, SoE EN SoI (SOURCE HSO THE RESULT COMPANY).	62
FIGUUR 2-1, CONTINUOUS DESIGN PYRAMID.	65
FIGUUR 2-2, NON CONTINUOUS DESIGN PYRAMID.	66
FIGUUR 4-1, VERANDERPARADIGMA.	71
FIGUUR 4-2, VERANDERPARADIGMA - BEELDVORMING.	72
FIGUUR 4-3, VERANDERPARADIGMA - MACHTSVERHOUDING.	74
FIGUUR 4-4, VERANDERPARADIGMA - ORGANISATIE.	77
FIGUUR 4-5, VERANDERPARADIGMA - RESOURCES.	78
FIGUUR 4-6, INGEVULD VERANDERPARADIGMA VOOR CONTINUOUS DESIGN KARAKTERISTIEKEN.	80
FIGUUR 4-7, INGEVULD VERANDERPARADIGMA VOOR WATERVAL DESIGN KARAKTERISTIEKEN.	80
FIGUUR 5-1, CONTINUOUS DESIGN PYRAMID MET DELIVERABLES EN TE BEANTWOORDEN VRAGEN.	85

FIGUUR 5-2, CONTINUOUS DESIGN PLANNING MODEL.	86
FIGUUR 6-1, CONTINUOUS DESIGN VALUE STREAM.	87
FIGUUR 6-2, USE CASE DIAGRAM VOOR CONTINUOUS DESIGN.	88
FIGUUR 7-1, BUSINESS VIEW.	93
FIGUUR 7-2, SYSTEM CONTEXT DIAGRAM TEMPLATE.	94
FIGUUR 7-3, SYSTEM CONTEXT DIAGRAM VOORBEELD.	95
FIGUUR 7-4, VALUE STREAM CANVAS TEMPLATE.	97
FIGUUR 7-5, VALUE STREAM CANVAS VOORBEELD.	99
FIGUUR 8-1, SOLUTION VIEW.	101
FIGUUR 8-2, USE CASE DIAGRAM BASIS TEMPLATE.	102
FIGUUR 8-3, USE CASE DIAGRAM TEMPLATE MET EEN INCLUDE USE CASE.	103
FIGUUR 8-4, USE CASE DIAGRAM TEMPLATE MET EXTEND USE CASE.	103
FIGUUR 8-5, 16-VLAKSMODEL.	104
FIGUUR 8-6, VOORBEELD USE CASE DIAGRAM 'KOFFIESERVICE'.	105
FIGUUR 8-7, TEMPLATE SYSTEM BUILDING BLOCKS – ALGEMEEN.	106
FIGUUR 8-8, TEMPLATE SYSTEM BUILDING BLOCKS – INFORMATIE.	108
FIGUUR 8-9, TEMPLATE SYSTEM BUILDING BLOCKS – APPLICATIE.	110
FIGUUR 8-10, TEMPLATE SYSTEM BUILDING BLOCKS – TECHNOLOGIE.	113
FIGUUR 8-11, VOORBEELD VAN INFORMATION SYSTEM BUILDING BLOCKS VAN EEN KOFFIESERVICE.	115
FIGUUR 8-12, VOORBEELD VAN APPLICATION SYSTEM BUILDING BLOCKS VAN EEN KOFFIESERVICE.	115
FIGUUR 8-13, VOORBEELD VAN TECHNOLOGY SYSTEM BUILDING BLOCKS VAN EEN KOFFIESERVICE.	116
FIGUUR 8-14, VALUE STREAM MAPPING TEMPLATE.	117
FIGUUR 8-15, VOORBEELD-1 VALUE STREAM MAPPING KOFFIESERVICE.	118
FIGUUR 8-16, VOORBEELD-2 VALUE STREAM MAPPING KOFFIESERVICE.	118
FIGUUR 9-1, DESIGN VIEW.	119
FIGUUR 9-2, TEMPLATE USE CASE SCENARIO.	125
FIGUUR 9-3, VOORBEELD VAN TWEE USE CASE SCENARIO'S.	126
FIGUUR 10-1, REQUIREMENT VIEW.	127
FIGUUR 10-2, BDD TEMPLATE.	129
FIGUUR 10-3, GHERKIN FEATUREFILE VOORBEELD.	130
FIGUUR 11-1, TEST VIEW.	131
FIGUUR 12-1, CODE VIEW.	135
FIGUUR 13-1, BE-MORE VOOR EEN FSA BIJ ASSURITAS.	145
FIGUUR 13-2, FSA BEELDVORMING BIJ ASSURITAS.	146
FIGUUR 13-3, DE HUIDIGE EN DE GEWENSTE DOCUMENTATIE WERKWIJZE BIJ ASSURITAS.	146
FIGUUR 13-4, FSA MACHT BIJ ASSURITAS.	147
FIGUUR 13-5, FSA ORGANISATIE BIJ ASSURITAS.	148
FIGUUR 13-6, IMPLEMENTATION CONTINUOUS DESIGN AT ASSURITAS.	148
FIGUUR 13-7, FSA TEMPLATE.	149
FIGUUR 13-8, FSA USAGE IN THE DEVELOPMENT PROCESS.	149
FIGUUR 13-9, FSA RESOURCES.	150

DEEL III. CONTINUOUS TESTING

FIGUUR 4-1, VERANDERPARADIGMA.	161
FIGUUR 4-2, VERANDERPARADIGMA - BEELDVORMING.	162
FIGUUR 4-3, VERANDERPARADIGMA- MACHTSVERHOUDING.	164
FIGUUR 4-4, VERANDERPARADIGMA - ORGANISATIE.	168
FIGUUR 4-5, VERANDERPARADIGMA - RESOURCES.	170
FIGUUR 5-1, IDEAL TEST PYRAMID.	176

FIGUUR 5-2, NON-IDEAL TEST PYRAMID.	177
FIGUUR 6-1, VALUE STREAM VOOR CONTINUOUS TESTING.	183
FIGUUR 6-2, USE CASE DIAGRAM VOOR CONTINUOUS TESTING.	185
FIGUUR 7-1, VOORBEELD UNITTESTEN.	195

DEEL IV. CONTINUOUS INTEGRATION

FIGUUR 2-1, VAN SOURCECODE NAAR BINARYCODE.	203
FIGUUR 2-2, HET BOUWEN VAN EEN OPLOSSING.	204
FIGUUR 2-3, AGILE PROGRAMMEREN.	205
FIGUUR 2-4, BASISBEGRIPPEN VAN CONTINUOUS INTEGRATION.	205
FIGUUR 4-1, VERANDERPARADIGMA.	213
FIGUUR 4-2, VERANDERPARADIGMA - BEELDVORMING.	214
FIGUUR 4-3, VERANDERPARADIGMA- MACHTSVERHOUDING.	217
FIGUUR 4-4, VERANDERPARADIGMA - ORGANISATIE.	220
FIGUUR 4-5, VERANDERPARADIGMA - RESOURCES.	222
FIGUUR 5-1, KLASSENMODEL VERSIEBEHEER.	228
FIGUUR 5-2, LOKALE BACK-UP VERSIEBEHEER.	229
FIGUUR 5-3, LOKAAL MANUEEL VERSIEBEHEER.	229
FIGUUR 5-4, LOKAAL VERSIEBEHEERSYSTEEM.	230
FIGUUR 5-5, CENTRAAL VERSIEBEHEERSYSTEEM.	231
FIGUUR 5-6, GEDISTRIBUEERD VERSIEBEHEERSYSTEEM.	232
FIGUUR 6-1, CONTINUOUS INTEGRATION VALUESTREAM.	237
FIGUUR 6-2, USE CASE DIAGRAM VOOR CONTINUOUS INTEGRATION.	238
FIGUUR 7-1, GIT TOOLS.	246
FIGUUR 7-2, GIT VERSIONING.	246
FIGUUR 7-3, VOORBEELD VAN INCREMENTEEL EN ITERATIEF PROGRAMMEREN.	253

Tabellen

TABEL 1-1, CONTINUOUS EVERYTHING ASPECTEN.	2
TABEL 1-2, BIJLAGEN.	3

DEEL I. CONTINUOUS PLANNING

TABEL 2-1, PLANNINGSOBJECTEN.	13
TABEL 3-1, VEEL VOORKOMENDE PROBLEMEN BIJ HET HANTEREN VAN EEN PLANNINGSTOOLS.	18
TABEL 6-1, USE CASE TEMPLATE.	37
TABEL 6-2, USE CASE VOOR CONTINUOUS PLANNING.	40
TABEL 7-1, PI'S VAN DE SCOREKAART VAN OIRSOUW.	50
TABEL 7-2, ALTERNATIEVEN VAN PROGRAMMABESTURING.	51
TABEL 7-3, TOELICHTING PI'S VAN DE SCOREKAART VAN OIRSOUW.	52
TABEL 7-4, BSC - STRATEGISCHE PROJECT MANAGEMENT PRESTATIE INDICATOREN.	53
TABEL 7-5, VOORBEELD TOEPASSING VAN DE SCORECARD VAN OIRSOUW.	53
TABEL 7-6, VOORBEELD VAN EEN ROADMAP.	54
TABEL 7-7, TEMPLATE VAN EEN EPIC ONE PAGER.	54
TABEL 7-8, TEMPLATE VAN EEN EPIC ONE PAGER.	57

DEEL II. CONTINUOUS DESIGN

TABEL 3-1, VEEL VOORKOMENDE PROBLEMEN BIJ HET HANTEREN VAN EEN DESIGN.	70
TABEL 4-1, VERSCHILLEN TUSSEN EEN WATERVAL DESIGN EN EEN CONTINUOUS DESIGN.	79
TABEL 6-1, USE CASE TEMPLATE.	89
TABEL 6-2, USE CASE VOOR CONTINUOUS DESIGN.	92
TABEL 9-1, USE CASE NARRATIVE TEMPLATE.	122
TABEL 9-2, VOORBEELD VAN EEN USE CASE NARRATIVE.	125
TABEL 11-1, PYTHON UNITTEST TEMPLATE.	133
TABEL 11-2, PYTHON UNITTEST VOORBEELD.	133
TABEL 12-1, PYTHON HEADER ANNOTATIE VOORBEELD.	140
TABEL 12-2, C++ QT STYLE SCRIPT ANNOTATIEVOORBEELD.	141
TABEL 12-3, APPLICATIEDOCUMENTATIE GEGENEREERD DOOR DOXYGEN.	142

DEEL III. CONTINUOUS TESTING

TABEL 3-1, VEEL VOORKOMENDE PROBLEMEN BIJ HET TOEPASSEN VAN TESTEN.	160
TABEL 5-1, TESTSOORT-MATRIX TEMPLATE.	178
TABEL 5-2, TESTSOORT-MATRIX VOORBEELD.	178
TABEL 5-3, TESTTECHNIEK-MATRIX TEMPLATE.	179
TABEL 5-4, TESTTECHNIEK-MATRIX VOORBEELD.	180
TABEL 5-5, TESTOBJECT-MATRIX.	180
TABEL 5-6, TESTOBJECT-MATRIX VOORBEELD.	181
TABEL 5-7, TESTTOOL PATTERN.	181
TABEL 5-8, TESTTOOL VOORBEELD.	182
TABEL 6-1, USE CASE TEMPLATE.	186
TABEL 6-2, USE CASE VOOR CONTINUOUS TESTING.	189
TABEL 7-1, GHERKIN KEYWORDS.	192
TABEL 7-2, GHERKIN FEATUREFILE VOORBEELD.	192
TABEL 7-3, PYTHON UNITTEST TEMPLATE.	193
TABEL 7-4, PYTHON UNITTEST VOORBEELD.	194
TABEL 7-5, TESTSTRATEGIE TEMPLATE.	196
TABEL 7-6, TESTSTRATEGIE VOORBEELD.	196

DEEL IV. CONTINUOUS INTEGRATION

TABEL 3-1, VEEL VOORKOMENDE PROBLEMEN BIJ HET TOEPASSEN VAN SOFTWARE-ONTWIKKELING.	210
TABEL 5-1, FEATURES VAN VERSIEBEHEERSYSTEMEN.	233
TABEL 5-2, OVERIGE FEATURES VAN VERSIEBEHEERSYSTEMEN.	233
TABEL 5-3, RISICO'S VAN VERSIEBEHEERSYSTEMEN.	234
TABEL 5-4, EEN OVERZICHT VAN ENKELE VERSIEBEHEERSYSTEMEN.	235
TABEL 6-1, USE CASE TEMPLATE.	239
TABEL 6-2, USE CASE DIAGRAM VOOR CONTINUOUS INTEGRATION.	241
TABEL 7-1, STAPPEN OM EEN REPOSITORY AAN TE MAKEN.	247
TABEL 7-2, TOELICHTING OP DE STAPPEN OM EEN REPOSITORY AAN TE MAKEN.	247

Bijlagen

BIJLAGE A, LITERATUURLIJST.	257
BIJLAGE B, BEGRIPPENLIJST.	261

BIJLAGE C, AFKORTINGEN.....	277
BIJLAGE D, WEBSITES	281
BIJLAGE E, INDEX	283

Ten geleide

Development & Operations, kortweg DevOps is het uitgangspunt geweest om tot uitdieping te komen in Continuous Everything. Dit met een referentie aan de begrippen Continuous Integration/ Continuous Deployment (CI/CD) die veelvuldig aan de orde komen bij het concept DevOps. De aspecten van DevOps zijn gerelateerd aan het begrip Continuous en de stappen in de ontwikkel/beheercyclus (ook bekend als de DevOps Lemniscaat).

Doorgronding van DevOps houdt bedrijven bezig om een optimale invulling te verzorgen van de 'oude' concepten ontwikkeling en beheer. Helaas is er geen eenduidige uitwerking van DevOps te vinden in de literatuur c.q. op het grote Internet. Al snel wordt hier teruggegrepen op dat DevOps 'een filosofie' is. Met andere woorden: niet strak gedefinieerd en op meerdere wijzen uitlegbaar en invulbaar. Bedrijven worstelen derhalve met dit concept. Het concept Continuous Everything geeft een eenvoudige en uniforme structuur om de kennis en knowhow van elk Continuous Everything-aspect zoals Continuous Integration en Continuous Deployment te definiëren.

Dit boek 'Continuous Development' omvat vier aspecten van Continuous Everything, namelijk Continuous Planning, Continuous Design, Continuous Testing en Continuous Integration. Dit vormt een dik boek, waarin een stukje kennis en ervaring wordt ontsloten dat Bart de Best heeft vergaard op het gebied van Continuous Development.

In dit boek is een zeer uitvoerige beschrijving van deze Continuous Everything aspecten van DevOps opgenomen. Daarbij horen de diverse best practices die vanuit praktijkervaring worden aangedragen in een theoretische context. Deze context maakt het mogelijk om de Continuous Everything aspecten aan elkaar te relateren.

Met gepaste trots hebben we met een kleine groep professionals Bart ondersteund bij de totstandkoming van de uitwerking van alle aspecten van Continuous Everything. Met de niet te stoppen drive van Bart ligt hier nu een heel gevulde gereedschapskist met best practices voor DevOps. Met name de samenhang is hierin een gedegen aanvulling voor het hanteren van de begrippen rondom de aspecten van DevOps.

Veel leesplezier, bladerend door het boek, beschouwend op Continuous Everything!

Dr. Louis van Hemmen – BitAll b.v.

Voorwoord

Dit boek is samengesteld op basis van mijn ervaringen met Continuous Everything. Dit begrip duidt twee aspecten die de wereld van DevOps (Development & Operations) kenschetsen te weten 'Continuous' en 'Everything'. Het continue karakter van DevOps is vooral terug te vinden in de hoge frequentie van de levering en de fast feedback die hierdoor verkregen wordt. Everything duidt op het gegeven dat niet alleen software Continuous moet worden geleverd maar dat alle aspecten van de informatisering hierin mee moeten bewegen.

In dit boek staat continue ontwikkeling centraal. Het doel van dit Continuous Everything-gebied is het beheersen van alle value streams gerelateerd aan de ontwikkeling en het onderhoud van applicaties om de tijdige realisatie van de beoogde outcome van de business value streams te borgen en daarmee de bedrijfsdoelen te behalen. Dit boek is een bundel van vier CE-boeken, namelijk Continuous Planning, Continuous Design, Continuous Testing en Continuous Integration.

Dit 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 hoe om te gaan met dit onderdeel van Continuous Everything, heb ik besloten dit boek Agile te houden. Dit houdt in dat ik in dit boek elk aspect beknopt beschrijf. Ik deel hierbij belangrijke inzichten, die ik heb opgedaan gedurende mijn rol als consultant, trainer, coach en examiner ten aanzien van die werkgebied. 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. Tevens heb ik de kennis en kunde vertaald naar diverse trainingen die ik verzorg. Deze zijn te vinden op 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 | www.nielstalens.nl |
| • 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 Everything 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.

Bart de Best, Zoetermeer.

bartb@dbmetrics.nl

1 Inleiding

Leeswijzer:

De eerste paragraaf van dit hoofdstuk geeft het doel weer van dit boek (1.1). Daarna wordt de doelgroep (1.2) benoemd. Paragraaf 1.3 bespreekt achtergrond van Continuous Development en paragraaf 1.4 de structuur en de inhoud van het boek door per deel in het kort aan te geven wat wordt behandeld. Dit hoofdstuk sluit af met een leeswijzer (1.5).

1.1 Doelstelling

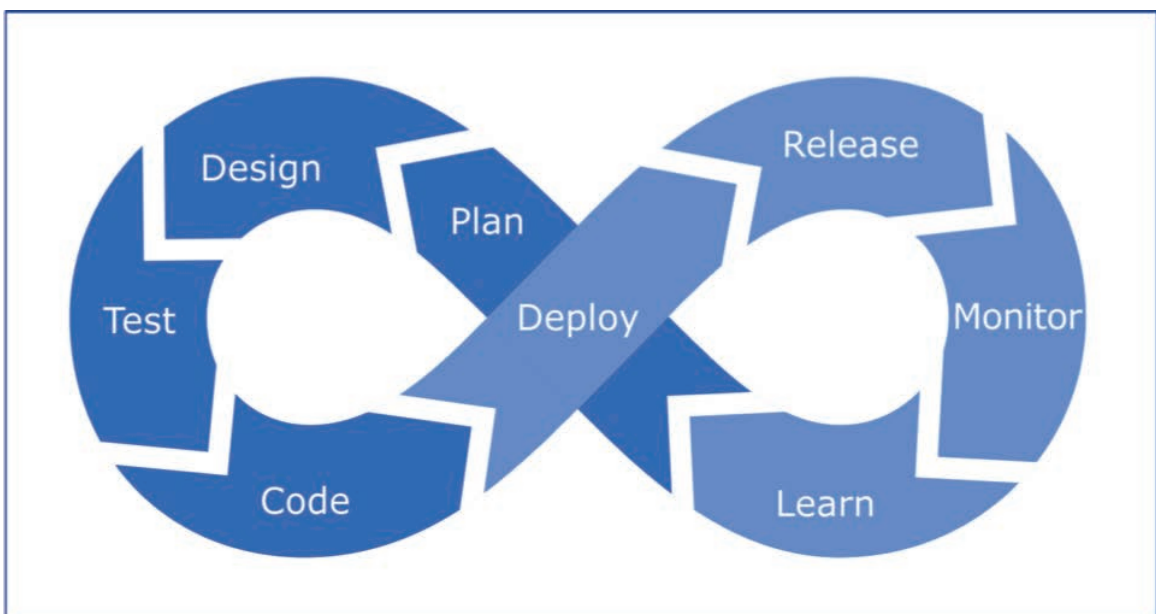
De primaire doelstelling van dit boek is het bieden van een Continuous Development gereedschapskist. Dit boek bespreekt vier belangrijke Continuous Development aspectgebieden. Er zijn zeker nog vele andere aspectgebieden, maar de in dit boek geselecteerde zijn een goede basis voor Continuous Development. De diepgang van de aspectgebieden is beperkt gehouden vanwege de beperkte publicatieruimte. Het boek is bedoeld als naslagwerk voor iedereen die met DevOps te maken heeft.

1.2 Doelgroep

De doelgroep van Continuous Development zijn alle betrokken functionarissen 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. Ten slotte is er een doelgroep die niet ontwikkelt of beheert maar die vaststelt of de value streams voldoen aan de vereiste criteria. Deze doelgroep omvat kwaliteits-medewerkers 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 methoden technieken om op een continue wijze invulling te geven aan Continuous Development. De DevOps Lemniscaat geeft een overzicht van de belangrijkste Continuous Development (donker blauw) en Continuous Operations (licht blauw) aspectgebieden, zoals weergegeven in [Figuur 1-1](#). Continuous Development en Continuous Operations vormen samen het Continuous Everything concept. Dit boek beschrijft alleen het Continuous Development gedeelte.



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 Everything (CE).

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 Robotics en Continuous Growth. Dit is gedaan om de DevOps Lemniscaat overzichtelijk te houden.

Het woord 'Continuous' duidt op een aantal karakteristieken die de werkzaamheden binnen een DevOps team duiden. Ten eerste is de frequentie van handelingen hoger dan bij traditioneel systeemontwikkeling. Dit heeft betrekking op zowel de bouw als het deployen van hetgeen is gebouwd. Dit kan variëren van minuten, uren en dagen in ontwikkelfrequentie. Daarnaast heeft 'Continuous' betrekking op het holistisch beschouwen van de werkzaamheden. Zo blijft het testen niet beperkt tot de ontwikkelomgeving maar worden testen uitgevoerd in alle omgevingen. Tevens worden niet alleen de producten en services getest maar ook valuestreams en zelfs mensen hun kennis en kunde. Dit sluit aan op de people, process, partner en technology views van ITIL 4. Ten slotte duidt de term 'Continuous' dat alle fasen van de DevOps Lemniscaat aan elkaar zijn gerelateerd. Zo wordt Continuous Testing gebruikt in de stappen 'Plan', 'Design', 'Code', 'Deploy' en 'Monitor'.

1.4 Structuur

Dit boek is opgebouwd door vier eerder gepubliceerde boeken samen te vatten in dit boek te weten:

- deel I DevOps Continuous Planning
- deel II DevOps Continuous Design
- deel III DevOps Continuous Testing
- deel IV DevOps Continuous Integration

1.4.1 Deel I, Continuous Planning

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.

Continuous Planning omvat de planningstechnieken 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.

1.4.2 Deel II, Continuous Design

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.

Continuous Design omvat het 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 en Continuous Documentation.

1.4.3 Deel III, Continuous Testing

Continuous Testing is een aanpak die beoogt om een snelle 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. Continuous Testing wordt in dit deel van het boek gedefinieerd 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, Business Driven Development, Test Driven Development, test policies, test technieken, test tools en rol van unit testcases in Continuous Testing.

1.4.4 Deel IV, Continuous Integration

Continuous Integration is een holistische Lean software ontwikkelaanpak die beoogt om op een incrementele en iteratieve wijze continue software te produceren en in productie te nemen waarbij waste reductie hoog in het vaandel staat.

Door de incrementele en iteratieve werkwijze van Continuous Integration wordt fast feedback mogelijk omdat functionaliteiten eerder in productie kunnen worden genomen. Hierdoor wordt waste gereduceerd omdat de herstelacties sneller zijn omdat deze eerder worden gevonden en sneller kunnen worden opgelost. Continuous Integration wordt besproken 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 build-in failure mode.

1.5 Bijlagen

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

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	Websites	Een aantal relevante websites zijn in deze bijlage opgenomen. In dit boek wordt verwezen naar deze websites door de referentie: [http Name].
E	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, www.scrumguides.org .
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 [Wiki]
A/B testing	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.
Acceptance test	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 (FAT), User Acceptance Testcases (UAT) and Production Acceptance Testcases (PAT). The FAT and UAT should be expressed in the language of the business.
Affinity	DevOps is about collaboration 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.
Agile Infrastructure	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.
Alternate path	See happy path .
Andon cord	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 [Kim 2016] .
Anomaly detection techniques	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"> • peer to peer programming; • demonstrating weekly progress; • documentation; 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"> • Culture: Culture relates to the people and process aspects of DevOps. Without the right culture, automation attempts will be fruitless. • Automation: Release management, configuration management, and monitoring and control tools should enable automation. • Measurement: 'If you can't measure it, you can't manage it.' & 'If you can't measure it, you can't improve it'. • Sharing: Culture of sharing ideas and problems is critical to help organisations to improve. Creates feedback loop.
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"> • The first act begins in IT Operations where technical debt results in jeopardising our most important organisational promises. • 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. • The third stage is where the deployments are getting slower and slower, and outages are increasing. The business value continuously decreases.
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"> • Independent: The product backlog item should be self-contained, in a way that there is no inherent dependency on another product backlog item. • Negotiable: Product backlog items, up until they are part of an iteration, can always be changed, rewritten or even discarded. • Valuable: Product backlog item must deliver value to the stakeholders.

Begrip	Betekenis
	<ul style="list-style-type: none"> • Estimable: The size of a product backlog item must always estimable. • Small: Product backlog items should not be so big as to become impossible to plan / task / prioritise with a certain level of certainty. • Testable: The product backlog item or its related description must provide the necessary information to make test development possible.
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"> • clear understanding of the goals; • understanding the right way to work; • ensure high quality of work; • getting the work right for 100% completion, never pass defects to the next process; • Definition of Done (DoD) is vital; 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"> • elimination of waste (<u>muda</u>'s); • <u>JIT</u>; • standardisation of production; • cycle of continuous improvements. <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"> • defining problems with supporting data; • making sure everybody recognises the problems clearly;

Begrip	Betekenis
	<ul style="list-style-type: none"> • setting a hypothesis on the problems found; • defining countermeasure actions to verify the hypothesis; • defining countermeasure actions be in daily based activities; • measuring a weekly KPI so people can feel a sense of accomplishment.
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"> • direction (target); • current condition (IST situation); • target condition (SOLL situation); • PDCA (Deming wheel). <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"> • A3 thinking (problem solving) • Continuous flow (eliminates waste)

Begrip	Betekenis
	<ul style="list-style-type: none"> • Kaizen • Kanban • KPI (Key Performance Indicator) • Plan Do Check Act (PDCA) • Root cause analysis • Specific, Measurable, Accountable, Realistic, Timely (SMART) • Value stream mapping (depict the flow) • JJK (No defects are passed to next process)
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 (ITSM) 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"> • Debug level: Information at this level is about anything that happens in the program, most often used during debugging. • Info level: Information at this level consists of actions that are user-driven or system specific. • Warn level: Information at this level tells us of conditions that could potentially become an error. • Error level: Information at this level focuses on error conditions. • Fatal level: Information at this level tells us when we must terminate.
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 [Wiki].</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"> • information management; • and on-the-spot decision making.
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"> • Functional: Functional-oriented organisations optimise for expertise, division of labour, or reducing cost. • Matrix: Matrix-oriented organisations attempt to combine functional and market orientation. • Market: Market-oriented organisations optimise for responding quickly to customer needs.
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"> • Pathological organisations are characterised by large amounts of fear and threat. • Bureaucratic organisations are characterised by rules and processes. • Generative organisations are characterised by actively seeking and sharing information to better enable the organisation to achieve its mission.

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"> • 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. • 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.
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"> • situational information to describe what is happening; • background information or context; • an assessment of what they believe the problem is; • recommendations for how to proceed.
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"> • Janitor Monkey (unused resources); • Chaos Monkey (try to shut down a service); • Conformity Monkey (non-conformance to rules).
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 σ 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 [Wiki] .
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 [Wiki].</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"> • Most of the errors are found using unit tests as early as possible. • 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. • Any errors should be found with the fastest possible category of testing.
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"> • Most of the investment is in manual and integration testing. • Errors are found later in the testing. • Slower running automated tests are performed first.
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 [Whatis].</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"> • The first way – flow understand and increase the flow of work (left to right); • The second way – feedback create short feedback loops that enable continuous improvement (right to left); • The third way – Continuous Experimentation and Learning (continuous learning).
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"> • Unnecessary software features. • Communication delays. • Slow application response times. • Overbearing bureaucratic processes.
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.

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

Tabel D-1, Websites.

Bijlage E, Index

%

%C/A · 47, 54, 277

A

A/B testing · 261

acceptatie

- criterium · 159, 191, 203, 204, 205, 209, 250
- omgeving · 157, 169, 173, 177, 184, 188, 197, 243
- test · 153, 177, 188, 201

acceptatiecriterium · 262, 266

acceptatietest · 261

account data · 112

actor · 36, 38, 88, 89, 90, 102, 121, 122, 123, 185, 186, 187, 238, 239, 240

ad hoc · 26

affinity · 261

AFM · 277

Agile · 25, 75, 165, 171, 191, 193, 195, 218, 261, 273, 274

- coach · 1
- design · 61
- design pyramid · 135
- infrastructure · 261
- principe · 18
- project · 7, 13, 21, 32, 48
- project portfolio · 49
- release train · 25, 75, 218
- Scrum · 74
- start architectuur · 46
- test management · 101
- werkwijze · 191

Agile Release Train · 165

Agile Scrum · 24, 37, 164, 166, 169, 216, 265

- aanpak · 160, 210
- framework · 24, 25, 74, 75, 165, 166, 216, 218
- guide · 18
- proces · 24, 74, 165, 216, 217
- team · 165, 217

Algemene Verordening

Gegevensbescherming · Zie AVG

alignment · 147

alternate path · 261

Amazon Web Services · Zie AWS

analyseerbaarheid · 220

Andon cord · 261

anker · 21, 71, 213

annotatie · 220, 221, 252

anomaly detection technique · 261

antifragility · 262

anti-pattern · 22, 23, 25, 27, 28, 65, 72, 73, 76, 77, 78, 79, 155, 163, 166, 169,

170, 173, 193, 196, 197, 215, 219, 221, 223, 253, 262

apenrots · 21, 72, 213

applicatie · 153, 156, 157, 162, 163, 165, 180, 181, 183, 186, 187, 188, 191, 193, 214, 217

- architectuur · 12
- component · 67, 102
- object · 168
- object-matrix · 168
- objecttype · 169
- portfolio · 52

applicatiebeheer · 268

applicatiecomponent · 269

application business rule · 138

architect · 1, 18, 23, 25, 218, 219, 251

architectuur · 7, 10, 12, 18, 21, 26, 27, 31, 32, 33, 34, 37, 38, 40, 41, 45, 46, 50, 66, 107, 109, 138, 143, 145, 147, 150, 153, 161, 166, 170, 201, 213, 218

- model · 12, 52, 170
- principe · 7, 12, 29, 30, 31, 33, 38, 46, 52, 145, 166, 170, 219
- view · 12

artefact · 35, 87, 153, 158, 184, 201, 237, 262, 264

artefact repository · 262

Artefactory · 206, 241

assessment · 28, 70, 89, 90, 272

Atomic operation · 233

auditor · 1

automated test · 262

automatiseringstools · 191

Autoriteit Financiële Markten · Zie AFM

availability · 263

AVG · 61, 277

AWS · 112, 277

Azure · 112

B

backlog item · 178, 179, 195, 271

back-up & recovery · 157

bad apple theory · 262

bad path · 262

Balanced Score Card · Zie BSC

base use case · 37, 89, 186, 239

baseline · 215

BDD · 63, 86, 127, 128, 129, 131, 132, 136, 137, 138, 150, 154, 179, 191, 195, 262, 277

BDD-aanpak · 86

bedrijfsdoel · 37, 50

bedrijfsproces · 50, 93

beeldvorming · 7, 21, 22, 23, 26, 27, 63, 71, 72, 73, 76, 78, 154, 201, 213, 214, 215, 216, 219, 222

Behavior Driven Development · Zie BDD
beheerbaarheid · 52

beheerder · 162, 164, 214
 beheerfunctie · 52
 beheerorganisatie · 50
 beheervoorziening · 157
 beleid · 114
 best practice · 21, 71, 154, 161, 165, 191, 195, 213, 217, 263
 best-of-breed tool · 170
 besturing · 65
 besturingsmodel · 29
 bevoegdheden · 63, 71, 201, 213, 227
 bevoegdheid · 7, 21
 BI · 277
 BI-model · 61
 binary · 201, 206, 207, 248, 262
 - code · 203, 206
 biometrieke · 114
 BI-oplossing · 62
 blackbox · 23, 114
 blackbox test · 156, 157, 186
 blameless post mortem · 262
 blamelessness · 262
 blue/green deployment · 262
 Body of Knowledge · Zie BOK
 Boehm · 155
 BOK · 277
 bottleneck · 25, 44
 bottom-up · 21, 71, 161, 167, 213, 219
 boundary · 30, 34, 41, 42, 44, 45
 bouwen · 171, 187
 bouwsteen · 46, 47, 67, 101, 105, 106, 107, 108, 109, 111, 114, 116, 117, 121, 125, 210, 216
 branch · 177
 branchen · 214
 branching · 263
 broken build · 262
 brown field · 262
 BSC · 277
 budget · 51
 build · 262, 263, 264, 273
 business
 - architect · 95
 - architectuur · 12
 - case · 9, 10, 17, 21, 26, 32, 37, 38, 39, 48, 163
 - information manager · 95
 - opportuniteit · 50
 - perspectief · 177
 - procesactiviteit · 52
 - requirement · 177
 - rule · 52
 - transactie · 112
 - value · 263, 265
 - value stream · 26, 76
 - view · 63, 65, 93
 business – IT alignment · 45
 Business Canvas Model · 98
 Business Intelligence · Zie BI
 Business Value System · Zie BVS
 BVS · 277
 bytecode · 204

C

C/A · 97
 CA · 49, 153, 201, 277
 CAB · 50, 147, 277
 cadans · 204
 CAMS · 264, 277
 canary releasing · 263
 capability · 264
 Capability Maturity Model Integration · Zie CMMI
 capaciteit · 263
 cascaderen · 51
 CCTA Risk Assessment Method
 Methodology · Zie CRAMM
 CD · 153, 154, 155, 186, 201, 263, 267, 277
 CE · 153, 201, 277
 CEM · 277
 CEMLI · 277
 Central Event Monitor · Zie CEM
 Central Processing Unit · Zie CPU
 CEO · 277
 certificaat · 112
 CFO · 277
 champion gedrag · 218
 Change Advisory Board · Zie CAB
 change category · 263
 change schedule · 263
 check-out · 240
 Chief Executive Officer · Zie CEO
 Chief Finance Officer · Zie CFO
 Chief Information Officer · Zie CIO
 Chief Technology Officer · Zie CTO
 CI · 154, 155, 186, 263, 267, 277
 CI/CD pipeline · 210, 245, 248
 CI/CD secure pipeline · 7, 17, 22, 24, 74, 78, 153, 155, 159, 162, 165, 170, 171, 176, 178, 186, 201, 202, 204, 209, 215, 216, 217, 219, 223, 240
 CIA · 277
 CIA-matrix · 157
 CIO · 277
 CI-tool · 245
 CL · 277
 classificatie · 231
 cloud · 221, 263
 - broker · 114
 - provider · 114
 cloud configuration file · 263
 cloud service · 263
 cluster immune system release pattern · 263
 CM · 277
 CMDB · 277
 CMMI · 277
 CMS · 277
 CN · 277
 CO · 277
 coachen · 217
 CoC · 277
 code branch · 263
 Code of Conduct · Zie CoC

- code review form · 263
 - code view · 63, 65
 - codestandaard · 165, 210, 215, 217, 230
 - codeverlies · 234
 - codified NFR · 263
 - collaboratie tool · 79, 95, 122
 - collaboration · 263
 - commandline · 133, 244
 - commit code · 263
 - commit stage · 263
 - communicatiemiddel · 46
 - communication service · 114
 - Communities of Practice · Zie CoP
 - competence · 261, 266
 - competentie · 52
 - Competitive Advantage · Zie CA
 - Competitive Response · Zie CR
 - Completeness / Accurateness · Zie %C/A
 - complexiteit · 51, 52
 - compliance · 264
 - compliance checking · 264
 - compliance officer · 264
 - compliancey · 264
 - compliance officer · 264
 - component · 267, 270, 273
 - concept · 153
 - concurrent gebruikers · 157, 177
 - concurrentie · 49, 51
 - druk · 51
 - positie · 50
 - timing · 49, 50
 - voordeel · 49
 - confidentiality · 277
 - Confidentiality, Integrity & Availability · Zie CIA
 - configuratie · 234
 - fout · 234
 - Configuration Item · Zie CI
 - configuration management · 264
 - Configuration Management DataBase · Zie CMDB
 - Configuration Management System · Zie CMS
 - Configuration, Extention, Modification, Localisation, Integration · Zie CEMLI
 - Confluence · 77, 95, 122
 - constructiefout · 12
 - container · 264
 - continuity · 263, 269
 - Continuous
 - Assessment · 2, 17, 18, 22, 23, 24, 29, 30, 31, 36, 69, 70, 72, 73, 74, 83, 85, 88, 89
 - Auditing · 2
 - Deployment · 2
 - Design · 2
 - Design Pyramid · 21, 65, 71, 77, 83, 85, 86, 93, 101, 109, 119, 127, 131, 135, 143
 - Documentation · 63, 135, 136, 137, 138
 - Everything · 1, 7, 17, 61, 69, 153, 160, 201, 210
 - Improvement · 274
 - Integration · 2, 213
 - Learning · 2, 274
 - Monitoring · 2
 - Monitoring Layer model · 29
 - Planning · 2, 37
 - Testing · 2, 153, 154, 201, 202
 - Testing roadmap · 18, 25, 70, 75, 160, 165, 211, 217
 - Continuous aSessment · Zie CS
 - Continuous Auditing · Zie CA
 - Continuous Deployment · Zie CD
 - Continuous design · Zie CN
 - 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 · 26, 28, 45, 264, 272
 - Conway's law · 264
 - CoP · 25, 75, 165, 166, 218, 219, 277
 - core value stream · 7, 17, 22
 - counter measure · 268
 - CP · 201, 277
 - CPU · 277
 - consumptie · 227
 - cycle · 220
 - CR · 49, 277
 - CRAMM · 277
 - CRC · 277
 - CS · 277
 - CT · 153, 277
 - CTO · 277
 - Cucumber · 128, 132, 191
 - cultural debt · 264
 - Culture, Automation, Measurement and Sharing · Zie CAMS
 - current state · 96, 97, 98
 - CY · 277
 - cycle time · 30, 38, 264
 - Cyclic Redundancy Check · Zie CRC
-
- D**
- data
 - analyse · 52
 - base · 156, 203, 209
 - storage · 114
 - storage service · 114
 - debt · 264
 - debuggen · 209
 - declarative programming · 264
 - deelproduct · 10, 25, 56
 - defect · 159, 160, 162, 163, 168, 169, 173, 177, 184, 187, 215, 221, 222, 243, 244, 248, 249, 269
 - reductie · 177
 - defect tracking · 265
 - definitie · 7, 153, 201
 - Definition of Done · Zie DoD

Definition of Ready · Zie DoR
 Definitional Uncertainty · Zie DU
 Definitive Media Library · Zie DML
 dekkingsgraad · 27, 28, 163, 180
 deliverable · 63, 72, 79, 86, 93, 101, 119, 127, 131, 135
 Demming wheel · 268
 deployen · 154, 155, 158
 deployment · 261

- pipeline · 63, 110, 112, 113, 116, 117, 122, 131, 132, 135, 136, 137
- pipeline tool · 112

 deployment pipeline · 263
 design · 10, 11, 21, 34, 37, 55, 61, 264, 275
 design view · 63, 65, 121
 desinvestering · 50
 Dev engineer · 1
 developer · 157, 165, 167, 170, 180
 development · 7, 17, 24, 29, 56, 57, 261, 262, 265, 267, 268, 270, 272, 273, 274, 275

- team · 24, 25
- value stream · 22

 Development & Operations · Zie DevOps
 development ritual · 265
 development team · 74, 75, 165, 166, 167, 216, 217, 218, 219
 Development Value System · Zie DVS
 Development, Test, Acceptance and Production · Zie DTAP
 DevOps · 21, 24, 62, 71, 72, 74, 153, 159, 160, 161, 162, 164, 166, 167, 169, 170, 179, 184, 187, 201, 209, 210, 213, 214, 216, 218, 219, 223, 261, 263, 265, 271, 275, 277

- architect · 166, 218, 219
- capabilities · 70
- capability · 18
- coach · 145
- engineer · 17, 18, 23, 28, 30, 31, 34, 218, 219, 223, 253, 254, 261, 263, 264, 265, 272, 273, 274
- engineer · 253
- engineer · 253
- engineers · 76
- Lemniscaat · 1, 2, 61
- organisatie · 170
- team · 1, 23, 24, 25, 27, 38, 74, 75, 147, 179, 261, 262, 263, 264, 266, 270, 275
- werkwijze · 1

 DML · 278
 DMZ service · 113
 DNS · 278
 DNS service · 113
 documentatie · 193
 documentatie management · 112
 documentatiegenerator · 135, 136, 139
 DoD · 37, 38, 58, 61, 67, 89, 90, 132, 137, 160, 206, 223, 239, 267, 278
 doelstelling · 7, 61, 153, 201
 Domain Name System · Zie DNS

doorlooptijd · 227
 DoR · 206, 278
 downward spiral · 265
 driver · 197, 253
 DTAP · 272, 278
 DTAP environments · 272
 DU · 50, 278
 duurzaam · 44, 221
 DVS · 15, 42, 278

E

E2E · 278
 e-commerce · 16
 editors · 209, 216
 een technical debt · 219
 eenduidige taal · 31
 effectief · 26, 74, 165, 217
 efficiënt · 74, 165, 217
 eigenaarschap · 21, 24, 25, 70, 71, 74, 76, 160, 161, 164, 165, 166, 167, 211, 213, 216, 217, 218, 219
 e-mail pass around · 265
 emerging architectuur · 218
 emerging design · 187
 enable value stream · 9, 26, 29
 enabling services · 46
 enabling value stream · 7, 17, 22
 End User eXperience Monitoring · Zie EUX
 End-to-End · Zie E2E
 enterprise

- architect · 38, 49
- architectuur · 67, 117
- architectuurmodel · 12, 46
- business rule · 138

 Enterprise Resource Planning · Zie ERP
 Enterprise Service Buss · Zie ESB
 Entity Relation Diagram · Zie ERD
 epic · 11, 13, 22, 26, 30, 31, 32, 34, 39, 40, 53, 54, 55, 56, 57, 72, 79, 86, 98, 101, 120, 127, 165, 166, 217, 218, 278
 epic eigenaar · 54
 epic one pager · 91, 210
 Epic Solution Approach · Zie ESA
 epic-ontwerp · 147, 148
 ERD · 278
 ERP · 278
 error guessing · 179
 error path · 265
 ESA · 145, 147, 278
 ESB · 278
 E-shaped · 266
 E-shaped people · 214, 215, 219, 223, 225, 226
 ETL · 278
 EUX · 278
 event · 270
 executable · 204
 eXtensible Markup Language · Zie XML
 Extract Transform & Load · Zie ETL
 eXtreme Programming · Zie XP

F

failure · 262
 fast feedback · 21, 30, 71, 143, 159, 160, 161, 163, 173, 176, 210, 213, 235
 FAT · 155, 156, 169, 177, 178, 179, 180, 181, 182, 188, 195, 261, 278
 feature · 11, 13, 32, 34, 55, 57, 79, 86, 98, 120, 122, 127, 128, 129, 131, 136, 137, 138, 145, 150, 192, 265, 266, 275
 Feature Solution Approach · Zie FSA
 feature toggle · 265
 feature-ontwerp · 148
 federaal model · 23
 feedback · 72, 80, 81, 105, 144, 177, 207, 210, 215, 226, 227, 244, 264, 265, 268, 271, 274
 feedforward · 265
 file system · 156, 194
 financiële baten · 51
 firmware · 159
 first time right · 97
 five times why methode · 18, 70, 160, 211
 flow · 66, 93, 94, 95, 98, 102, 126, 139, 264, 267, 268, 269, 270, 271, 272, 274, 275
 fout · 169, 176, 196, 253
 foutafhandeling · 251
 fragiliteit · 222
 framework · 203, 209, 251, 270
 front-end applicatie · 24, 74, 217
 FSA · 145, 146, 147, 149, 150, 278
 functie · 25, 28, 75, 156, 165, 166, 170, 186, 187, 193, 194, 217, 218
 functie analyse · 52
 functionaleiteits · 155
 functioneel beheerder · 169
 functioneel ontwerp · 66
 functionele acceptatiecriteria · 54
 Functionele AcceptatieTest · Zie FAT
 functionele requirement · 163

G

Galbraith · 75
 Gartner · 51
 GAT · 157, 278
 Gaussian distribution · 261, 265
 GCC · 61, 278
 GDPR · 278
 gebruiker · 197
 Gebruiker AcceptatieTest · Zie GAT
 gebruikersorganisatie · 1, 12
 gedragsprobleem · 221
 gedragsspecificatie · 191
 Gene Kim · 265, 270, 274
 General Computer Controls · Zie GCC
 General Data Protection Regulation · Zie GDPR
 Generieke & Specifieke Acceptatiecriteria · Zie GSA

gereedschapskist · 1, 151
 gewenste gedrag · 69
 gewenste situatie · 12, 27, 45, 46
 Gherkin · 77, 128, 129, 131, 132, 136, 137, 155, 183, 191, 192
 Gherkin language · 31, 67, 85, 128, 129, 132, 137
 Gherkin taal · 77
 Git · 147, 206, 234, 243, 244, 245, 246, 247, 253
 GIT · 197, 278
 Git repository · 244
 GitHub · 244, 245, 246, 247, 253
 GitLab · 244, 245
 Given When Then · 266, Zie GWT
 Global Information Tracker · Zie GIT
 goal · 11, 36, 37
 golden plated · 79
 governance · 25, 75, 166, 167, 218, 219
 Grady Booch · 157
 Graphical User Interface · Zie GUI
 green build · 202, 221, 222, 239, 243, 248
 green code · 209, 210, 227
 green field · 266
 GSA · 278
 GUI · 156, 157, 168, 181, 203, 244, 278
 guild · 166, 219
 GWT · 67, 86, 90, 91, 128, 132, 139, 155, 177, 191, 192, 266, 278

H

handmatige testcase · 177
 hand-off · 226
 Hand-off Readiness Review · Zie HRR
 happy flow · 35, 87, 237
 happy path · 179, 261, 262, 266
 hardware · 155, 159, 169, 264, 267, 275
 her aankopen · 43, 44
 herbruikbaarheid · 220
 hergebruik · 132
 hero · 215
 hero gedrag · 218
 Hero gedrag · 166
 herstellkosten · 215, 248
 holistisch · 22, 159, 162, 210, 214
 holistische aanpak · 155, 159
 Holocracy · 266
 homoniem · 22, 160, 162, 214
 horizontal splitting of feature · 266, 275
 HRM · 28, 29, 61, 83, 221, 278
 HRM beleid · 28
 HRM functionaris · 28
 HRR · 278
 huidige situatie · 10, 12, 18, 27, 45, 46
 Human Resource Management · Zie HRM

I

IaC · 63, 136, 214, 249, 261, 267, 278
 ICT · 267, 278

ID · 278
 ideal test pyramid · 154, 168, 169, 173, 176, 177, 274
 idempotent · 266
 Identifier · Zie ID
 IIR · 51
 impact · 50, 51, 62
 impactanalyse · 95
 imparative programming · 266
 inchecken · 196, 227, 253
 incident · 163
 incrementeel · 153, 164, 183, 201, 204, 205, 226, 235, 240, 251, 252, 253
 incrementeel ontwikkelen · 204, 252
 incrementeel programmeren plaats · 204
 Independent, Negotiable, Valuable, Estimatable, Small and Testable · Zie INVEST
 informatie

- planning · 50
- planningsproject · 49
- systeem · 1, 16, 17, 18, 23, 24, 46, 50, 52, 74, 153, 165, 166, 169, 191, 192, 216, 217, 218
- voorziening · 16, 50, 51

 Information assets, People, Organisation, Products and services, Systems and processes · Zie IPOPS
 Information Communication Technology · Zie ICT
 information radiator · 267
 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 · 267
 Infrastructure as Code · Zie IaC
 infrastructure component · 267
 infrastructure management · 267
 Infrastructure Risk · Zie IR
 infrastructuur

- architectuur · 12
- monitortool · 113
- portfolio · 52

 inhaalinvestering · 51
 input · 203, 215, 226, 251
 integratie monitorkennis · 28
 integrity · 277
 intermediair · 143
 International Standard On Assurance Engagements · Zie ISAE
 INVEST · 266, 278
 investering · 49
 IP address · 267
 IPOPS · 278
 IR · 50, 278

I-shaped · 266
 I-shaped people · 167, 170, 171, 215, 219, 223
 ISMS · 278
 ISO · 278
 isolatie · 207
 IST · 268
 Ist situatie · 18, 27, 37, 38, 144
 ISVS · 15, 42, 278
 IT · 61, 86, 107, 109, 265, 269, 274, 278
 iteratief · 153, 164, 183, 201, 204, 205, 226, 235, 240, 251, 252, 253
 ITIL · 51
 ITIL 4 · 278
 ITSM · 269, 278

J

Java Virtual Machine · Zie JVM
 Ji-Kotei-Kanketsu · Zie, Zie JKK
 JIT · 267, 268, 278
 JKK · 267, 278
 JSON · 136
 Just In Time · Zie JIT
 JVM · 204, 278

K

Kaizen · 243, 248, 249, 267, 269
 Kaizen Blitz (or Improvement Blitz) · 268
 Kaizen in advance · 268
 Kanban · 268, 269, 275
 kanban board · 57
 Ken Schwaber · 24, 74, 165, 216
 kennis · 156, 159, 162, 167, 168, 170, 197, 209, 214, 219, 223, 253, 254
 kennis domein · 171
 kennis overdracht · 18, 61, 70, 166, 218
 keten · 24, 74, 165, 166, 183, 217, 218
 Key Performance Indicator · Zie KPI
 kibana dashboard · 268
 klanttevredenheid · 11, 44
 kostenpost · 220
 kostprijs · 163, 170
 KPI · 42, 44, 54, 268, 269, 275, 278
 Kritieke Succes Factor · Zie KSF
 KSF · 42, 43, 44, 278
 kwalitatieve requirement · 163
 kwaliteit · 75, 153, 155, 166, 169, 196, 197, 218, 253, 254
 kwaliteits

- eis · 155, 156, 169
- medewerkers · 1

L

lagenstructuur · 86
 LAN · 244, 279
 landelijke voorziening · 221

Laravel · 209, 251
 late feedback · 154, 163, 169, 173, 202, 235
 latent defect · 268
 Launch Readiness Review · Zie LRR
 launching guidance · 268
 LCM · 279
 LDAP · 279
 LDAP service · 113
 Lead Time · 268, Zie LT
 Lean · 159, 210, 221, 248, 274, 275

- indicator · 14
- Six Sigma · 67, 97, 117
- testaanpak · 159
- tool · 268

 learning culture · 269
 leesbaarheid · 23
 lemniscaat · 153, 201
 levenscyclus · 30, 84
 library · 209
 lifecycle · 155, 171, 265, 267
 LifeCycle Management · Zie LCM
 Lightweight Directory Access Protocol · Zie LDAP
 lijnmanager · 1
 limitation · 26, 30, 34, 41, 42, 44, 45
 Local Area Network · Zie LAN
 log · 270
 logging level · 269
 long lived branch · 207, 233, 244
 loosely coupled architecture · 269
 loosely coupled services · 269
 loosely coupled systems · 218
 LRR · 268, 279
 LT · 47, 54, 97, 117, 123, 268, 279

M

machinetaal · 204
 machtsverhouding · 7, 21, 23, 24, 25, 26, 27, 63, 71, 73, 74, 76, 78, 154, 201, 213, 216, 217, 219, 222
 main · 206, 247
 mainline · 206, 207
 Management Information · Zie MI
 manuele taak · 163
 manufacturing process · 275
 marker · 37, 67, 90, 91
 marketing plan · 39
 marktsegment · 98
 MASR · 279
 master · 206, 217
 mastertestplan · 163, 183
 MD · 156
 Mean Time Between Failure · Zie MTBF
 Mean Time Between System Incidents · Zie MTBSI
 Mean Time To Repair · Zie MTTR
 medewerker · 50, 51
 meetinstrument · 51
 merge hell · 177, 234

mergen · 22, 214, 220, 221, 235, 244, 246, 251
 meta-data · 262
 metadata · 55, 57, 72, 121, 138, 163, 169, 215, 216, 220, 221, 226
 metadatabeheer · 221
 methodiek · 153
 MFA · 279
 MI · 49, 279
 microservice · 135, 137, 269
 microservice architecture · 269
 Microsoft Operations Framework · Zie MOF
 migratiepad · 27
 milestone · 227
 mini pipeline · 269
 Minimal Viable Product · Zie MVP
 Minimum Required Information · Zie MRI
 modelleringswijze · 52
 Modify, Avoid, Share, Retain · Zie MASR
 modulair programmeren · 137
 modulariteit · 221
 module · 156, 180, 181, 186
 Module Test · Zie MT
 MOF · 279
 monitor

- matrix · 28
- model · 7
- voorziening · 7, 18, 23, 25, 26, 153

 monitoring · 270
 monolithic · 270
 MRI · 269, 279
 MT · 156, 184, 279
 MTBF · 279
 MTBSI · 279
 MTTR · 270, 279
 muda · 270
 Multi Factor Authentication · Zie MFA
 multidisciplinair · 209
 multiple expertise · 170, 223
 MVP · 13, 31, 39, 54, 56, 101, 221, 279

N

navigator · 197, 253
 NC · 279
 netwerkcommunicatie · 113
 netwerkverbinding · 231, 233, 244
 NFR · 116, 263, 270, 279
 Non Conformity · Zie NC
 non Continuous

- Design Pyramid · 66

 Non Functional Requirement · Zie NFR
 non-ideal test pyramid · 173, 177, 225
 non-SQL database · 112
 nulmeting · 65

O

OAWOW · 279
 obeya · 270
 object code · 262

Oirsouw · 49
 OLA · 279
 onafhankelijkheid · 27
 One Agile Way of Working · Zie OAWOW
 one piece flow · 270
 one-delivery-pipeline · 114
 ontwerp · 12, 147, 148, 149
 ontwerper · 187
 ontwerptechniek · 77
 Ontwikkel-, Test-, Acceptatie- en Productieomgeving · Zie OTAP
 ontwikkelaar · 162, 164, 166, 167, 194
 ontwikkelomgeving · 155, 156, 157, 159, 160, 162, 163, 168, 169, 173, 177, 184, 186, 187, 194, 201, 215, 244, 251
 ontwikkelpad · 28
 ontwikkelproces · 24, 74, 138, 193, 216, 220
 onzekerheidsreductie principe · 75
 oorzaak · 18, 26, 44, 45, 154, 201
 open source · 233
 operating system · 233
 Operational Level Agreement · Zie OLA
 operational requirement · 250
 operationeel doel · 51
 operationeel niveau · 50
 operationele manager · 50
 operations · 7, 17, 29, 261, 265, 270, 272, 275
 operations story · 270
 operations value stream · 22
 operator · 187
 Ops engineer · 1
 Ops liaison · 270
 organisatie · 51
 organisatie doel · 51
 organisatie vormgeving · 7, 63, 154, 201
 organisation archetype · 270
 organisational typology model · 270
 OTAP · 279
 OTAP-straat · 29, 158, 168
 outcome · 9, 17, 25, 26, 30, 42, 45, 54, 171, 192
 output · 203
 output stroom · 66, 93, 94, 102
 over-the-shoulder · 271

P

PAAS · 49, 54, 114, 279
 package · 271
 pair programming · 197, 253, 254, 263, 271
 paperless · 163
 paperware · 159
 PAT · 157, 177, 178, 179, 180, 181, 182, 195, 261, 279
 patch · 62
 pattern · 146, 147, 149, 178, 193, 196, 197, 225, 228, 231, 234, 250, 251, 252, 253, 262, 271

PBI · 168, 173, 177, 178, 180, 181, 182, 279
 PDCA · 268, 269, 279
 peer review · 271
 peer to peer programming · 263
 peer-to-peer · 244
 PEN-test · 197
 People, Process & Technology · Zie PPT
 performance · 36, 38, 177, 263, 269, 275, 279
 Performance StressTest · Zie PST
 performancecriterium · 36, 88, 185, 238
 personalisatie · 112
 PESTLE · 279
 PI scorekaart Oirsouw · 50
 piekbelasting · 215
 pipeline · 261, 267, 269, 272, 273, 275
 Plan Do Check Act · Zie PDCA
 planningscyclus · 18
 planningsobject · 11, 13, 17, 18, 23, 29, 30, 31, 32, 34, 40, 55, 177, 178
 Platform As A Service · Zie PAAS
 Plug-in IDE · 233
 policy · 154
 Political, Economic, Sociological, Technological, Legislative, Environmental · Zie PESTLE
 POP · 28, 29
 POR · 50, 279
 post mortem · 271
 post-condition · 121
 PPT · 7, 17, 29, 61, 69, 83, 107, 153, 155, 159, 203, 209, 210, 279
 pre-condition · 121, 128, 192
 prestatie indicatoren · 117
 principe · 22, 72, 162, 214
 probleem · 17, 18, 19, 69, 70, 154, 159, 160, 201, 209, 210, 211
 proces
 - analyse · 52
 - beschrijving · 51
 - eigenaar · 1
 - manager · 1
 - plan · 10, 55, 56
 proceseigenaarschap · 165, 217
 Processing Time · Zie PT
 product
 - backlog · 29, 39, 55, 56, 57, 72, 73, 75, 86, 112, 120, 177, 178, 195, 271
 - backlog item · 25, 75, 267
 - owner · 1, 18, 23, 24, 30, 38, 39, 54, 55, 57, 58, 216, 271
 - roadmap · 9, 10, 32, 37, 39, 42
 - vision · 10, 26, 27, 32, 33, 34, 37, 39, 40, 41, 42, 48, 49, 53, 55
 Product Backlog Item · Zie PBI
 Productie AcceptatieTest · Zie PAT
 productieomgeving · 138, 155, 163, 177, 184, 243, 244
 productieproces · 11
 production environment · 269
 programmabesturing · 51
 programmeerstijl · 196, 253

programmeertaal · 203, 209, 227, 235, 250
 programmeren · 202, 203, 204, 205, 209, 214, 215, 220, 221, 250, 252, 253
 programmeur · 153, 156, 194, 196, 197
 programming paradigm · 271
 project · 49, 50, 51, 52
 - deliverable · 52
 - doelstelling · 50
 - leiding · 50
 - medewerker · 51
 Project or Organisational Risk · Zie POR
 promoten · 155
 PSA · 52
 pseudocode · 204, 220, 252
 PSQL · 264
 PST · 157, 177, 178, 179, 180, 181, 182, 195, 279
 PT · 47, 54, 97, 117, 123, 279
 pull request · 138, 249, 253
 pull request process · 271
 pull-request · 197
 Python · 132, 133, 134, 138, 139, 193

Q

QA · 271, 279
 QC · 279
 Quality Assurance · Zie QA
 Quality Control · Zie QC

R

RACI · 25, 75, 166, 218, 279
 RASCI · 25, 75, 166, 218, 279
 RBAC · 279
 RBAC service · 113
 Real User Monitoring · Zie RUM
 recursiebeginsel · 15
 reduce batch size · 271
 reduce number of handoffs · 271
 redundantie · 23, 73, 163, 215
 refactoring · 184, 188, 193
 referentiearchitectuur · 38, 52, 147, 166, 219
 refinement · 72
 refinieren · 120
 regressietest · 163, 177, 186, 188, 194
 regressietestbasis · 160
 release · 2, 62, 271
 release manager · 271
 release pattern · 271
 rendement · 49
 repository · 193, 202, 206, 207, 216, 219, 223, 230, 231, 232, 233, 240, 241, 243, 244, 247, 262, 263, 271, 272
 REpresentational State Transfer Application Programming Interface · Zie REST API
 requirement · 12, 18, 23, 31, 34, 39, 47, 63, 66, 73, 77, 79, 86, 102, 114, 119, 122, 127, 128, 129, 131, 135, 136, 138,

139, 147, 150, 153, 155, 159, 162, 163, 169, 171, 177, 183, 186, 187, 191, 203, 209, 215, 243, 262, 268, 270, 273, 279
 requirement view · 63, 65
 research · 1
 resource · 7, 22, 50, 63, 72, 161, 201, 214
 Resources · 154
 Responsibility, Accountable, Consulted and Informed · Zie RACI
 Responsibility, Accountable, Supporting, Consulted and Informed · Zie RASCI
 REST API · 279
 REST-API · 156, 157, 168, 181
 retrospective · 39, 265
 Return On Investment · Zie ROI
 review · 265
 risico · 1, 52, 62, 145, 160, 178, 180, 181, 191, 193, 195, 196, 197, 210, 215, 216, 234, 243, 248, 249, 250, 251, 253, 262, 268
 risicoanalyse · 31, 32, 38, 42, 48, 95, 157, 183, 221
 risicoanalysesessie · 47
 risico-eigenaar · 216
 risk · 50
 risk based planning · 27, 31
 ritme · 204
 roadmap · 26, 71, 165, 213, 217, 218, 219, 226
 roadmap to value · 7, 9, 18, 32, 33
 ROI · 49, 51, 52, 53, 215, 279
 rol · 25, 28, 35, 75, 87
 Role-Based Access Control · Zie RBAC
 rootcause · 17, 69, 159, 209
 rootcause analyse · 269
 RUM · 279

S

SA · 50, 279
 sad path · 271
 SAFe · 279
 SAFe framework · 25, 26, 75, 76, 166, 218
 safety check · 271
 Sarbanes Oxley · Zie SoX
 SAT · 157, 177, 178, 179, 180, 181, 182, 195, 279
 SBAR · 272, 279
 SBB · 37, 47, 63, 89, 101, 102, 105, 106, 116, 122, 186, 239, 280
 SBB-A · 280
 SBB-I · 106, 107, 108, 109, 111, 112, 114, 118, 122, 123, 124, 130, 192, 280
 SBB-I diagram · 106, 107, 108, 111
 SBB-T · 36, 37, 47, 89, 106, 112, 114, 118, 122, 125, 186, 239, 280
 Scaled Agile Framework · Zie SAFe
 S-CI · 279
 scorekaart · 11, 44
 scorekaart Oirsouw · 49, 50, 51
 - competitive advantage · 49
 - competitive respons · 49

- definitional uncertainty · 50
- infrastructure risk · 50
- management information · 49
- project or organisational risk · 50
- Return On Investment · 49
- strategic IS architecture · 50
- strategic match · 49
- technical uncertainty · 50
- scrum master · 165, 166, 216, 217, 218
- Scrum master · 1, 24, 25, 74, 75, 216
- Secure Sockets Layer · Zie SSL
- security · 263, 264, 270, 271, 272, 274
- Security Acceptatie Test · Zie SAT
- security baseline · 221
- security officer · 264
- security value stream · 22
- self service capability · 272
- service · 52, 279
 - management · 61
 - organisatie · 217, 251
 - portfolio · 52
 - request · 144
- Service Level Agreement · Zie SLA
- Service Value System · Zie SVS
- shared goals · 272
- Sharepoint · 221
- shift-left · 21, 71, 213, 226, 243
- shift-left organisatie · 132, 144, 176
- short lived branch · 207, 220, 233
- silo · 275
- Simian army · 272, 274
- Simple Network Management Protocol · Zie SNMP
- SIT · 156, 177, 178, 179, 180, 181, 182, 184, 195, 280
- Situation, Background, Assessment, Recommendation · Zie SBAR
- skill · 28, 29, 78, 79, 83
- skill matrix · 78, 79
- skills · 266
- SLA · 280
- SLA-norm · 44
- SM · 49, 280
- SMART · 269, 280
- SMART goal · 34
- SME · 25, 75, 94, 166, 218, 280
- smoke testing · 272
- SNMP · 280
- SoA · 280
- SoE · 16, 24, 61, 62, 74, 165, 166, 216, 217, 218, 273, 280
- software · 73, 155, 157, 159, 163, 164, 166, 179, 186, 191, 218, 262, 273, 275
 - conflict · 220
 - item · 119
 - lifecycle · 159, 209
 - ontwikkelaar · 164
 - ontwikkeling · 193
 - ontwikkelproces · 127, 155, 159, 162, 215, 216, 248, 249
- Software Configuration Item · Zie S-CI
- SoI · 61, 273, 280
- SOLL · 268
- Soll situatie · 12, 18, 27, 37, 38, 46, 144
- solution view · 63, 65, 93, 101
- SoR · 16, 24, 61, 62, 74, 79, 112, 165, 166, 183, 217, 218, 273, 280, Zie
- sourcecode · 63, 66, 67, 104, 131, 132, 133, 135, 136, 137, 138, 139, 147, 149, 159, 162, 164, 179, 191, 193, 196, 197, 220, 253, 254, 262, 263, 265, 272, 273, 275
- sourcecode header · 164
- sourcecodestandaard · 164
- SoX · 280
- SoX-wetgeving · 61
- Specific, Measurable, Accountable, Realistic, Timely · Zie SMART
- Spotify · 25, 75, 166, 218
- Spotify model · 25, 75, 144
- sprint · 9, 10, 13, 18, 25, 30, 32, 37, 39, 41, 55, 56, 57, 58, 265
 - backlog · 177, 178, 180, 181
 - tijd · 167
- sprint execution · 265
- sprint planning · 265
- SQL · 280
- SQL-injectie · 222
- SQL-statement · 222
- SRG · 206, 250, 280
- SSL · 280
- ST · 156, 163, 184, 280
- stakeholder · 10, 13, 32, 38, 39, 48, 53, 54, 61, 63, 66, 67, 69, 83, 86, 90, 93, 94, 95, 96, 98, 102, 103, 119, 128, 136, 145, 147, 266
- standard deviation · 272
- standard operations · 272
- Standard Rules & Guidelines · Zie SRG
- stand-up · 265
- Statement of Applicability · Zie SoA
- static analysis · 273
- story · 13, 32, 57, 79, 98, 127
- Strategic IS Architecture · Zie SA
- Strategic Match · Zie SM
- strategie · 7, 9, 10, 11, 12, 17, 18, 23, 25, 26, 27, 30, 31, 33, 34, 37, 38, 40, 42, 43, 44, 49, 50, 53, 153, 201
- strategiemonitor · 11, 18
- strategische
 - beleid · 51
 - doel · 51
 - positie · 50
- strategische niveau · 50
- Structured Query Language · Zie SQL
- stuurmogelijkheid · 50
- Subject Matter Expert · Zie SME
- super use case · 37, 89, 186, 239
- SVS · 14, 15, 16, 42, 280
- synoniem · 22, 160, 162, 214
- stelsel
 - ontwikkeling · 61
 - test · 156, 168, 220
- 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 Context Diagram · 63, 66, 89, 90, 93, 94, 95, 96, 98, 102
 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 · 61, 117, 120, 261
 taakanalyse · 216
 taalconstructie · 220, 227
 taalelement · 203, 227, 251
 Taken, Verantwoordelijkheden en Bevoegdheden · Zie TVB
 Target Operating Model · Zie TOM
 task · 13, 32, 57, 267
 TCO · 220, 280
 TCP · 280
 TDD · 63, 67, 131, 132, 136, 137, 138, 150, 153, 154, 159, 177, 179, 186, 187, 191, 193, 195, 197, 201, 222, 235, 239, 244, 251, 273, 280
 Team Foundation Server · Zie TFS
 technical debt · 71, 213, 218, 219, 248, 264, 265
 technical debt backlog · 71, 161, 166, 167, 213, 218, 219
 Technical Information Security Officer · Zie TISO
 Technical Uncertainty · Zie TU
 technisch ontwerp · 66, 131
 technology adaption curve · 273
 technology executive · 273
 tegenmaatregel · 27, 31, 32, 38, 42, 43, 45, 54
 template · 36, 37, 88, 89, 178, 179, 180, 181, 185, 186, 193, 196, 238, 239
 test · 75, 153, 155, 156, 157, 159, 162, 163, 164, 165, 166, 168, 169, 171, 173, 177, 178, 179, 180, 181, 183, 184, 186, 187, 188, 191, 193, 194, 195, 197, 201, 203, 204, 214, 215, 216, 218, 222, 223, 230, 235, 239, 240, 243
 - aanpak · 155
 - afdeling · 215
 - architectuur · 160, 210
 - automation · 170, 197, 244
 - basis · 155, 156, 157, 183, 186, 187
 - case · 63, 81, 114, 119, 122, 131, 132, 133, 153, 155, 159, 160, 162, 163, 169, 171, 177, 179, 183, 184, 187, 188, 191, 193, 197, 201, 209, 261, 262, 263
 - cyclus · 187
 - engineer · 215
 - framework · 132
 - harness · 273
 - inspanning · 131, 132
 - management · 153, 155, 156, 159, 160, 183, 211
 - object · 180, 181, 187, 197
 - object-matrix · 173, 180, 181
 - omgeving · 156, 157
 - pattern · 168, 197
 - situatie · 197
 - skills · 160, 210
 - soort · 154, 155, 156, 157, 159, 160, 163, 168, 169, 173, 177, 178, 180, 181, 183, 184, 187, 195, 197, 202
 - soorten · 154
 - soort-matrix · 168, 177, 178, 195
 - strategie · 154, 155, 168, 178, 180, 183, 186, 191, 195, 196
 - strategie pattern · 168
 - taal · 160
 - techniek · 178, 179, 180, 181, 195
 - techniek-matrix · 173, 178, 179, 180, 181, 195
 - tool · 154, 161, 169, 170, 181
 - view · 63, 65
 Test Driven Development · Zie TDD
 tester · 170, 187, 265
 testtechniek · 154, 155, 157, 160, 168, 173
 - API-testing · 196
 - branch testing · 196
 - code-driven testing · 196
 - error-handling testing · 196
 - interface testing · 196
 - loop testing · 196
 - negative testing · 196
 - static testing · 196
 TFS · 122, 280
 The Agile Manifesto · 273
 the ideal testing automation pyramid · 274
 The Lean movement · 274
 the non-ideal testing automation inverted pyramid · 274
 The Three Ways · 270, 274
 theme · 11, 13, 22, 26, 30, 31, 32, 34, 49, 72, 79, 86, 96, 98, 120, 127
 theory of constraints · 274
 tijd · 50
 tijdlijn · 9, 51
 Time To Market · Zie TTM
 time-to-market · 163, 164
 tips en trucks · 93, 96, 101, 105, 120, 128, 132, 136
 TISO · 280
 toetsen · 75, 153, 155, 166, 168, 186, 218
 TOM · 280
 tool
 - integratie · 170
 - leverancier · 170
 - portfolio · 166, 219
 tool-assisted code review · 275
 Total Cost of Ownership · Zie TCO
 Toyota Kata · 275
 Toyota Production System · Zie TPS

TPS · 14, 280
 traceerbaarheid · 70, 73, 79, 90, 104, 215, 216, 220, 223, 226
 Traceerbaarheid · 207
 trage feedback · 69
 transactie · 24
 transactieverwerking · 24, 74, 165, 216
 transformation team · 275
 Transmission Control Protocol · Zie TCP
 trunk · 202, 206, 207, 221, 244, 272
 T-shaped · 266
 T-shaped people · 170, 215, 223
 TSQL · 264
 TTM · 215, 280
 TU · 50, 280
 TVB · 219, 280

U

UAT · 280
 uitwijkvoorziening · 157
 UML · 120, 280
 Unified Modeling Language · Zie UML
 uniforme taal · 22, 23
 uniforme WoW · 23, 169
 uniformeren · 76
 Unit Test · Zie UT
 unittestcase · 90, 162, 163, 164, 168, 191, 193, 197
 Use Case · 35, 36, 37, 40, 63, 66, 67, 73, 87, 88, 89, 91, 92, 96, 97, 101, 102, 103, 104, 112, 116, 117, 119, 120, 121, 122, 123, 125, 126, 127, 138, 150, 154, 155, 183, 184, 185, 186, 187, 202, 237, 238, 239
 - Diagram · 7, 35, 36, 37, 63, 67, 87, 88, 89, 91, 101, 102, 103, 104, 116, 119, 150, 154, 155, 169, 183, 184, 186, 187, 202, 237, 239, 241
 - Model · 148
 - Narrative · 119, 120, 121, 122, 123, 125, 126
 - Scenario · 119, 120, 121, 125, 126
 - template · 37
 User Acceptance Test · Zie UAT
 User eXperience design · Zie UX design
 user interface test · 197
 userinterface · 96, 107, 209
 userstory · 11, 32, 34, 62, 86
 UT · 132, 133, 134, 136, 139, 154, 155, 156, 168, 173, 177, 184, 193, 194, 195, 252, 253, 280
 UX design · 280

V

vaardigheid · 50
 vakspecialisatie · 81
 value chain · 13, 14, 15, 16, 41, 42, 43, 44
 value chain doel · 42, 44

value stream · 1, 7, 9, 14, 15, 16, 17, 18, 22, 23, 26, 29, 30, 33, 34, 35, 37, 38, 42, 43, 44, 45, 46, 49, 54, 63, 66, 72, 84, 87, 89, 91, 93, 94, 95, 96, 97, 98, 101, 102, 104, 109, 114, 116, 117, 154, 155, 162, 170, 171, 183, 184, 202, 210, 214, 215, 226, 237, 240, 248, 249, 265, 269, 272, 273, 275, 280
 - canvas · 34, 42, 45, 63, 66, 89, 93, 94, 95, 96, 97, 98, 99, 102, 104, 109
 - canvas model · 63, 66, 93, 98, 104, 109
 - mapping · 89, 91
 - monitoring · 43, 46, 48, 49, 53, 55, 56, 57
 Value Stream Canvas · 90
 Value Stream Canvas model · 90
 Value Stream Mapping · Zie VSM
 VCR · 51, 280
 velocity · 167, 219, 220, 248, 249, 262
 veranderparadigma · 7, 21, 22, 23, 26, 27, 28, 63, 71, 72, 73, 76, 78, 143, 145, 154, 161, 162, 164, 167, 169, 201, 213, 214, 216, 219, 222
 verantwoordelijkheden · 63, 71, 201, 213, 227
 verantwoordelijkheid · 7, 21
 verbetering · 51
 versie · 62
 versiebeheer · 90, 95, 109, 122, 127, 132, 133, 135, 147, 206, 216, 217, 225, 227, 228, 229, 244, 246
 versiebeheersysteem · 228, 229, 230, 231, 232, 233, 234, 244, 252
 versiebeheertools · 213
 versieverlies · 234
 vertaalfout · 90
 vertical splitting of feature · 275
 Verwachte Contante ROI · Zie VCR
 verzekeringsmaatschappij · 143
 virtual machine · 113
 virtualised environment · 275
 visie · 7, 9, 11, 21, 22, 23, 28, 32, 33, 37, 38, 42, 43, 44, 72, 162, 214
 visualisatie · 22, 23, 275
 Voice over Internet Protocol · Zie VOIP
 VOIP · 280
 volwassen gedrag · 219
 volwassenheid · 74, 75, 76
 volwassenheidsmodel · 66
 volwassenwording · 165, 217
 voortbrengingsproces · 62, 226
 VSM · 63, 101, 116, 117, 275, 280

W

Wachttijd · 69
 walking skeleton · 275
 WAN · 280
 war room · 270
 waste · 18, 146, 155, 159, 162, 163, 164, 167, 169, 177, 195, 197, 207, 209, 210,

214, 215, 216, 226, 227, 248, 249, 251,
262, 264, 267, 268, 270, 274, 275
waste reductie · 155, 159, 162, 170, 207,
210, 214, 215, 275
waterval design · 73, 79, 80
watervalaanpak · 79, 80, 81
watervalproject · 61
Way of Working · Zie WoW
webbrowser · 244
webcontent management · 112
werkvermogen · 219
werkvoorraad · 17, 18, 55, 56
Westrum · 270, 271
wet & regelgeving · 206
whitebox test · 156, 186, 194
Wide Area Network · Zie WAN
Windows Management Instrumentation ·
Zie WMI
winst · 50
WIP · 280

WMI · 280
Work In Progress · Zie WIP
work item · 13
workflow · 264
WoW · 21, 24, 25, 58, 72, 159, 161, 164,
165, 167, 169, 213, 216, 217, 219, 280

X

XML · 280
XP · 131, 280

Z

Zachman · 12
zoekfaciliteit · 112

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

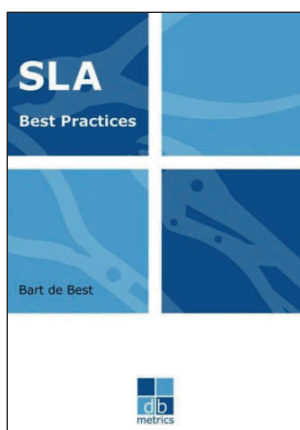
De eerste stap van een leven lang leren.

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

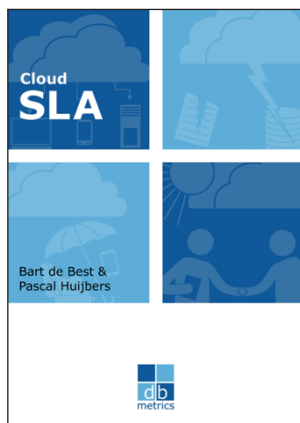
Het volledige ABC van service level agreements.

Het belangrijkste bij het leveren van een service is dat de klant tevreden is over de geleverde prestaties. Door deze tevredenheid verkrijgt de leverancier heraanbopen, 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

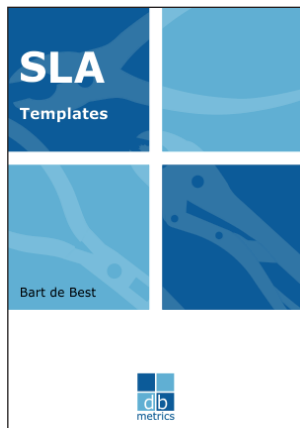
De best practices van cloud service level agreements

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 stringenter 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-ontwikkelproces en het beheren van de programmatuur die het Scrum-ontwikkelproces 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



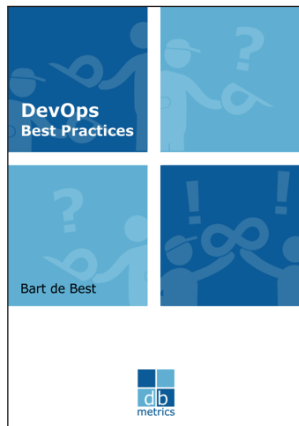
Agile Service Management met Scrum in de Praktijk

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

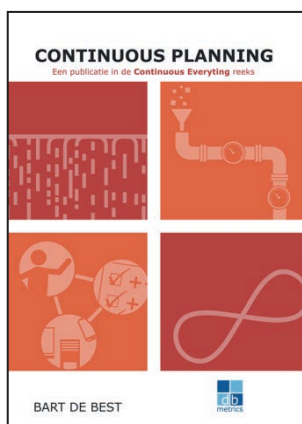
DevOps Architectuur Best Practices

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 continuous 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



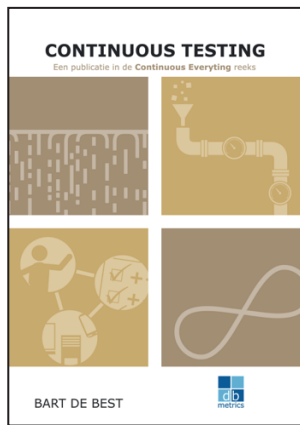
Continuous Design

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 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.

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 'continuous' 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 'continuous' 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

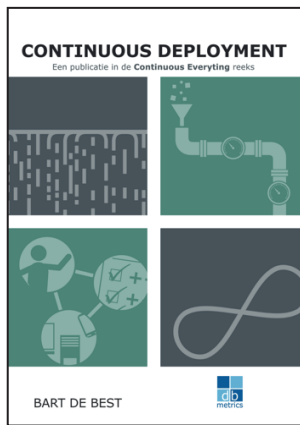
Een uitgave in de Continuous Everything reeks.

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

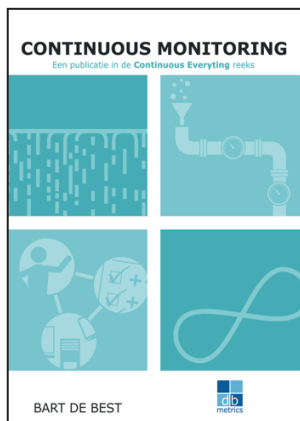
Een uitgave in de Continuous Everything reeks.

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

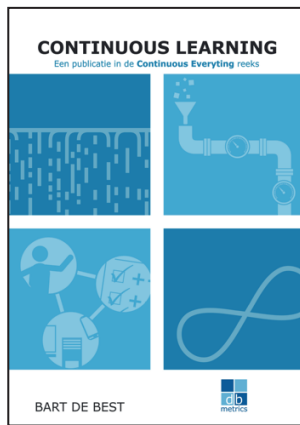
Een uitgave in de Continuous Everything reeks.

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.

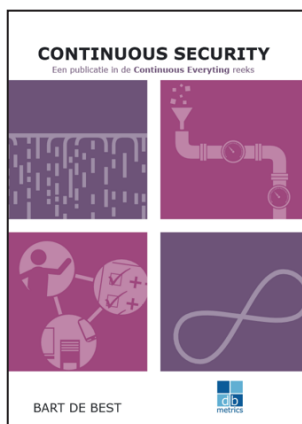
Continuous auditing is een aanpak die beoogt om DevOps teams in staat te stellen kort cyclisch aan te tonen in control te zijn bij het in hoog tempo realiseren, in productie nemen en beheren van de nieuwe of aangepaste producten en services.

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

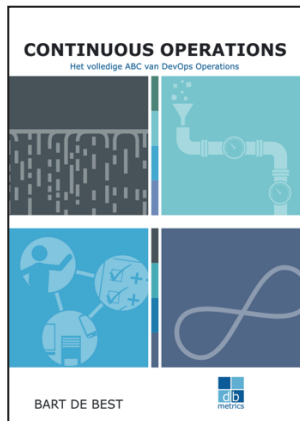
Een uitgave in de Continuous Everything reeks.

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

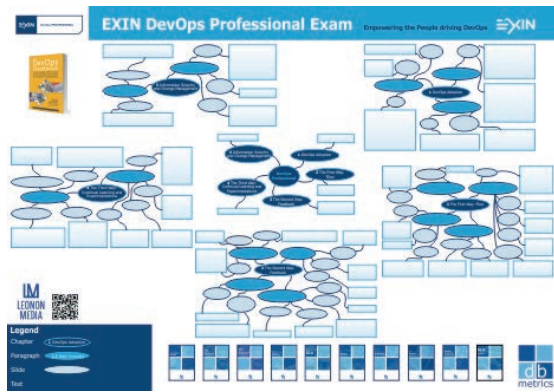
Een uitgave in de Continuous Everything reeks.

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.

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

CONTINUOUS OPERATIONS

Het volledige ABC van
DevOps Operations

Bart de Best



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 4 Continuous Everything boeken te weten:

1. Continuous Deployment
2. Continuous Monitoring
3. Continuous Learning
4. 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.

ISBN 978-94-92618-65-8

