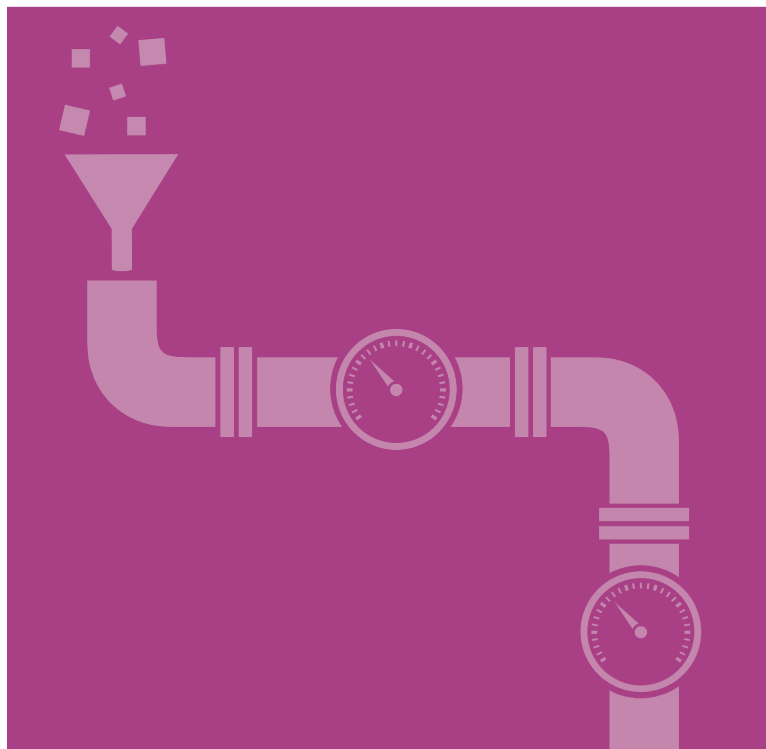


CONTINUOUS SECURITY

Een publicatie in de **Continuous Everything** reeks



BART DE BEST



DevOps Continuous Security Best Practices

Een uitgave in de Continuous Everything reeks

Bart de Best

Onder redactie van
Louis van Hemmen

Colofon

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

Leonon Media
(0)572 - 851 104

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

© 2022 Leonon Media

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

Titel : DevOps Continuous Security
Sub titel : Een uitgave in de Continuous Everything reeks
Datum : 15 december 2022
Auteur : Bart de Best
Uitgever : Leonon Media
ISBN13 : 978 94 91480 171
Druk : Eerste druk, derde editie 15 december 2022

© 2022, Leonon Media

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

TRADEMARK NOTICES

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

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

by Ellen Ullma

Inhoudsopgave

| | | |
|----------|---|-----------|
| 1 | INTRODUCTIE | 1 |
| 1.1 | DOEL | 1 |
| 1.2 | DOELGROEP | 1 |
| 1.3 | ACHTERGROND..... | 1 |
| 1.4 | STRUCTUUR | 4 |
| 1.4.1 | HOOFDSTUK 2: BASISCONCEPTEN EN BASISBEGRIPPEN | 4 |
| 1.4.2 | HOOFDSTUK 3: CONTINUOUS SECURITY DEFINITIE | 4 |
| 1.4.3 | HOOFDSTUK 4: CONTINUOUS SECURITY VERANKERING | 4 |
| 1.4.4 | HOOFDSTUK 5: CONTINUOUS SECURITY ARCHITECTUUR | 4 |
| 1.4.5 | HOOFDSTUK 6: CONTINUOUS SECURITY ONTWERP | 4 |
| 1.4.6 | HOOFDSTUK 7: CONTINUOUS SECURITY BEST PRACTICES | 4 |
| 1.4.7 | HOOFDSTUK 8: GOVERNANCE SECURITY PRACTICES | 4 |
| 1.4.8 | HOOFDSTUK 9: RISICO SECURITY PRACTICES | 4 |
| 1.4.9 | HOOFDSTUK 10: QUALITY SECURITY PRACTICES | 5 |
| 1.4.10 | HOOFDSTUK 11: CONTINUOUS SECURITY VERSUS AGILE SCRUM..... | 5 |
| 1.4.11 | HOOFDSTUK 12: CONTINUOUS SECURITY VERSUS DEVOPS | 5 |
| 1.4.12 | HOOFDSTUK 13: CONTINUOUS SECURITY ASSESSMENT | 5 |
| 1.5 | BIJLAGEN..... | 5 |
| 1.6 | LEESWIJZER..... | 5 |
| 2 | BASISCONCEPTEN EN BASISBEGRIPPEN | 7 |
| 2.1 | BASISCONCEPTEN..... | 7 |
| 2.1.1 | CONTINUOUS CONTROL | 7 |
| 2.1.2 | CONTINUOUS SECURITY PYRAMID | 7 |
| 2.1.3 | VALUE CHAIN..... | 9 |
| 2.1.4 | VALUE STREAM | 9 |
| 2.1.5 | DVS, SVS EN ISVS | 10 |
| 2.1.6 | POSITIONERING CONTINUOUS SECURITY | 11 |
| 2.1.7 | DE DRIE PERSPECTIEVEN VAN INFORMATION SECURITY | 12 |
| 2.2 | BASISBEGRIPPEN | 13 |
| 2.2.1 | RISICO MANAGEMENT BEGRIPPEN | 13 |
| 2.2.2 | VALUE SYSTEM BEGRIPPEN | 15 |
| 3 | CONTINUOUS SECURITY DEFINITIE | 17 |
| 3.1 | ACHTERGROND..... | 17 |
| 3.2 | DEFINITIE | 17 |
| 3.3 | TOEPASSING..... | 17 |
| 3.3.1 | OP TE LOSSEN PROBLEMEN..... | 17 |
| 3.3.2 | DE ROOT CAUSE | 18 |
| 4 | CONTINUOUS SECURITY VERANKERING..... | 21 |
| 4.1 | HET VERANDERPARADIGMA | 21 |
| 4.2 | BEELDVORMING..... | 22 |
| 4.2.1 | WAT WILLEN WE? | 22 |
| 4.2.2 | WAT WILLEN WE NIET?..... | 23 |
| 4.3 | MACHTSVERHOUDING | 23 |
| 4.3.1 | WAT WILLEN WE? | 24 |
| 4.3.2 | WAT WILLEN WE NIET?..... | 26 |
| 4.4 | ORGANISATIEVORMGEVING | 27 |
| 4.4.1 | WAT WILLEN WE? | 27 |
| 4.4.2 | WAT WILLEN WE NIET?..... | 28 |

| | | |
|----------|--|-----------|
| 4.5 | RESOURCES | 28 |
| 4.5.1 | WAT WILLEN WE? | 29 |
| 4.5.2 | WAT WILLEN WE NIET? | 29 |
| 5 | CONTINUOUS SECURITY ARCHITECTUUR..... | 31 |
| 5.1 | ARCHITECTUURPRINCIPES..... | 31 |
| 5.1.1 | ALGEMEEN | 31 |
| 5.1.2 | PEOPLE | 31 |
| 5.1.3 | PROCESS..... | 31 |
| 5.1.4 | TECHNOLOGY | 34 |
| 5.2 | ARCHITECTUURMODELLEN | 34 |
| 5.2.1 | CONTINUOUS SECURITY PYRAMID MODEL | 35 |
| 5.2.2 | CONTROL MODEL | 36 |
| 5.2.3 | QUALITY CONTROL & ASSURANCE MODEL | 37 |
| 5.2.4 | VALUE SYSTEMS..... | 38 |
| 5.2.5 | INFORMATION SECURITY VALUE SYSTEMS | 38 |
| 5.2.6 | SERVICE VALUE SYSTEMS..... | 41 |
| 5.2.7 | DEVELOPMENT VALUE SYSTEMS | 41 |
| 5.2.8 | INTEGRATED VALUE SYSTEMS | 42 |
| 6 | CONTINUOUS SECURITY ONTWERP..... | 45 |
| 6.1 | CONTINUOUS SECURITY VALUE STREAM | 45 |
| 6.2 | CONTINUOUS SECURITY USE CASE DIAGRAM | 46 |
| 6.3 | CONTINUOUS SECURITY USE CASE | 47 |
| 7 | CONTINUOUS SECURITY BEST PRACTICES..... | 53 |
| 7.1 | BEST PRACTICES | 53 |
| 7.2 | VALUE STREAM EXAMPLES..... | 53 |
| 8 | GOVERNANCE SECURITY PRACTICES | 55 |
| 8.1 | SCOPE GOVERNANCE SECURITY PRACTICES | 55 |
| 8.2 | VERKRIJG TOP MANAGEMENT COMMITMENT | 55 |
| 8.2.1 | USE CASE..... | 56 |
| 8.2.2 | DEFINITIE..... | 57 |
| 8.2.3 | DOELSTELLING | 57 |
| 8.2.4 | VOORBEELD | 57 |
| 8.2.5 | BEST PRACTICES | 58 |
| 8.3 | BEPAAL INTERESTED PARTIES | 58 |
| 8.3.1 | USE CASE..... | 58 |
| 8.3.2 | DEFINITIE..... | 59 |
| 8.3.3 | DOELSTELLING | 59 |
| 8.3.4 | VOORBEELD | 59 |
| 8.3.5 | BEST PRACTICES | 60 |
| 8.4 | BEPAAL SCOPE | 61 |
| 8.4.1 | USE CASE..... | 61 |
| 8.4.2 | DEFINITIE..... | 62 |
| 8.4.3 | DOELSTELLING | 62 |
| 8.4.4 | VOORBEELD | 62 |
| 8.5 | BEPAAL DOELEN..... | 63 |
| 8.5.1 | USE CASE..... | 63 |
| 8.5.2 | DEFINITIE..... | 64 |
| 8.5.3 | DOELSTELLING | 64 |
| 8.5.4 | VOORBEELD | 65 |

| | | |
|----------|---|-----------|
| 8.5.5 | BEST PRACTICES | 65 |
| 8.6 | BEPAAAL INFORMATION SECURITY POLICIES | 65 |
| 8.6.1 | USE CASE | 65 |
| 8.6.2 | DEFINITIE | 67 |
| 8.6.3 | DOELSTELLING | 67 |
| 8.6.4 | VOORBEELD INFORMATION SECURITY POLICY | 67 |
| 8.6.5 | VOORBEELD CODE OF CONDUCT | 67 |
| 8.6.6 | BEST PRACTICES DOELEN | 68 |
| 8.6.7 | BEST PRACTICES CODE OF CONDUCT | 68 |
| 9 | RISICO SECURITY PRACTICES | 69 |
| 9.1 | SCOPE RISICO SECURITY PRACTICES | 69 |
| 9.2 | BEPAAAL ISSUES - INTERNE | 70 |
| 9.2.1 | USE CASE INTERNAL ISSUES..... | 70 |
| 9.2.2 | DEFINITIE | 72 |
| 9.2.3 | DOELSTELLING | 72 |
| 9.2.4 | VOORBEELD..... | 72 |
| 9.2.5 | BEST PRACTICES – WOW IPOPS..... | 72 |
| 9.3 | BEPAAAL ISSUES - EXTERNAL | 75 |
| 9.3.1 | USE CASE EXTERNAL ISSUES | 75 |
| 9.3.2 | DEFINITIE | 76 |
| 9.3.3 | DOELSTELLING | 76 |
| 9.3.4 | VOORBEELD..... | 76 |
| 9.3.5 | BEST PRACTICES – WOW PESTLE..... | 76 |
| 9.4 | BEPAAAL CRAMM ISSUES | 79 |
| 9.4.1 | USE CASE CRAMM ISSUES | 79 |
| 9.4.2 | DEFINITIE | 79 |
| 9.4.3 | DOELSTELLING | 80 |
| 9.4.4 | VOORBEELD..... | 80 |
| 9.4.5 | BEST PRACTICES – WOW CRAMM..... | 81 |
| 9.5 | BEPAAAL RISK CRITERIA..... | 82 |
| 9.5.1 | USE CASE RISK CRITERIA | 82 |
| 9.5.2 | DEFINITIE | 84 |
| 9.5.3 | DOELSTELLING | 85 |
| 9.5.4 | VOORBEELD..... | 85 |
| 9.6 | BEPAAAL INFORMATION ASSETS | 91 |
| 9.6.1 | USE CASE INFORMATION ASSETS..... | 91 |
| 9.6.2 | DEFINITIE | 92 |
| 9.6.3 | DOELSTELLING | 92 |
| 9.6.4 | VOORBEELD..... | 92 |
| 9.7 | IDENTIFICEER RISICO'S..... | 93 |
| 9.7.1 | DEFINITIE | 94 |
| 9.7.2 | DOELSTELLING | 95 |
| 9.7.3 | VOORBEELD..... | 95 |
| 9.7.4 | BEST PRACTICES – WOW RISICO IDENTIFICATIE | 95 |
| 9.8 | VERRICHT RISK ASSESSMENT | 97 |
| 9.8.1 | DEFINITIE | 97 |
| 9.8.2 | DOELSTELLING | 97 |
| 9.8.3 | VOORBEELD..... | 98 |
| 9.8.4 | BEST PRACTICES – WOW RISK ASSESSMENT..... | 98 |
| 9.9 | VERRICHT RISK TREATMENT - OPTIONS | 99 |
| 9.9.1 | DEFINITIE | 100 |

| | | |
|-----------|--|------------|
| 9.9.2 | DOELSTELLING | 100 |
| 9.9.3 | VOORBEELD | 100 |
| 9.9.4 | BEST PRACTICES – WoW RISK TREATMENT | 100 |
| 9.10 | VERRICHT RISK TREATMENT - CONTROLS | 101 |
| 9.10.1 | DEFINITIE | 103 |
| 9.10.2 | DOELSTELLING..... | 103 |
| 9.10.3 | VOORBEELD | 103 |
| 9.10.4 | BEST PRACTICES – WoW RISK TREATMENT | 103 |
| 9.11 | VERRICHT RISK TREATMENT – BESTAANDE CONTROLS | 104 |
| 9.11.1 | DEFINITIE | 105 |
| 9.11.2 | DOELSTELLING..... | 105 |
| 9.11.3 | VOORBEELD | 105 |
| 9.12 | REALISEER CONTROLS | 105 |
| 9.12.1 | DEFINITIE | 107 |
| 9.12.2 | DOELSTELLING..... | 107 |
| 9.12.3 | VOORBEELD | 107 |
| 9.12.4 | BEST PRACTICES – WoW TREATMENT PLAN | 108 |
| 10 | QUALITY SECURITY PRACTICES | 109 |
| 10.1 | SCOPE QUALITY SECURITY PRACTICES..... | 109 |
| 10.2 | MONITOR EFFECTIVITEIT CONTROLS | 109 |
| 10.2.1 | DEFINITIE | 111 |
| 10.2.2 | DOELSTELLING..... | 111 |
| 10.2.3 | VOORBEELD | 111 |
| 10.2.4 | BEST PRACTICES – WoW MONITORVOORZIENING | 111 |
| 10.3 | VOER INTERNE AUDIT UIT - PLAN..... | 113 |
| 10.3.1 | DEFINITIE | 114 |
| 10.3.2 | DOELSTELLING..... | 114 |
| 10.3.3 | VOORBEELD | 114 |
| 10.3.4 | BEST PRACTICES – WoW INTERNE AUDIT | 114 |
| 10.4 | VOER INTERNE AUDIT UIT - CRITERIA..... | 115 |
| 10.4.1 | DEFINITIE | 116 |
| 10.4.2 | DOELSTELLING..... | 116 |
| 10.4.3 | VOORBEELD | 116 |
| 10.5 | VOER INTERNE AUDIT UIT - UITVOERING | 117 |
| 10.5.1 | DEFINITIE | 118 |
| 10.5.2 | DOELSTELLING..... | 118 |
| 10.5.3 | VOORBEELD | 118 |
| 10.6 | VOER INTERNE AUDIT UIT - RAPPORTAGE..... | 118 |
| 10.6.1 | DEFINITIE | 119 |
| 10.6.2 | DOELSTELLING..... | 119 |
| 10.6.3 | VOORBEELD | 119 |
| 10.7 | VERBETER CONTINUE - INCIDENTEN | 119 |
| 10.7.1 | DEFINITIE | 121 |
| 10.7.2 | DOELSTELLING..... | 121 |
| 10.7.3 | VOORBEELD | 122 |
| 10.8 | VERBETER CONTINUE – NON CONFORMITIES | 122 |
| 10.8.1 | DEFINITIE | 122 |
| 10.8.2 | DOELSTELLING..... | 122 |
| 10.8.3 | VOORBEELD | 123 |
| 10.9 | VERBETER CONTINUE – CSI..... | 123 |
| 10.9.1 | DEFINITIE | 124 |

| | | |
|-----------|---|------------|
| 10.9.2 | DOELSTELLING | 124 |
| 10.9.3 | VOORBEELD | 124 |
| 11 | CONTINUOUS SECURITY VERSUS AGILE SCRUM | 125 |
| 11.1 | POSITIONERING | 125 |
| 11.2 | AGILE MANIFESTO | 125 |
| 11.3 | AGILE METHODEN | 128 |
| 11.4 | AGILE SCRUM | 128 |
| 11.4.1 | DE AGILE SCRUM AANPAK | 129 |
| 11.4.2 | HET AGILE SCRUM ONTWIKKELPROCES PROCES..... | 129 |
| 11.4.3 | DE AGILE SCRUM TERMEN | 131 |
| 11.5 | CONTINUOUS SECURITY IN AGILE SCRUM | 133 |
| 11.6 | HET VERSCHIL | 136 |
| 12 | CONTINUOUS SECURITY VERSUS DEVOPS..... | 137 |
| 12.1 | DEVOPS POSITIONERING..... | 137 |
| 12.2 | DEVOPS CONCEPT | 137 |
| 12.2.1 | DEVOPS – DE OORSPRONG | 137 |
| 12.2.2 | DEVOPS – WAT IS HET? | 138 |
| 12.2.3 | DEVOPS GEVISUALISEERD | 138 |
| 12.2.4 | CONTINUOUS PLANNING..... | 139 |
| 12.2.5 | CONTINUOUS DESIGN..... | 139 |
| 12.2.6 | CONTINUOUS TESTING | 139 |
| 12.2.7 | CONTINUOUS INTEGRATION | 139 |
| 12.2.8 | CONTINUOUS DEPLOYMENT | 140 |
| 12.2.9 | CONTINUOUS MONITORING | 140 |
| 12.2.10 | CONTINUOUS LEARNING..... | 140 |
| 12.2.11 | CONTINUOUS ASSESSMENT | 140 |
| 12.3 | CONTINUOUS SECURITY IN DEVOPS | 140 |
| 12.4 | HET VERSCHIL | 142 |
| 13 | CONTINUOUS SECURITY ASSESSMENT | 145 |
| 13.1 | WAT IS HET CE-MODEL..... | 145 |
| 13.2 | VOLWASSENHEIDSDIMENSIES..... | 148 |
| 13.3 | DEVOPS CE MODEL, CY..... | 149 |
| | BIJLAGE A, LITERATUURLIJST | 155 |
| | BIJLAGE B, BEGRIPPENLIJST | 159 |
| | BIJLAGE C, AFKORTINGEN..... | 175 |
| | BIJLAGE D, WEBSITES | 179 |
| | BIJLAGE E, INDEX..... | 181 |

Figuren

| | |
|--|-----|
| FIGUUR 1-1, DEVOPS LEMNISCAAT. | 1 |
| FIGUUR 1-2, SOR, SOE EN SOI (SOURCE HSO THE RESULT COMPANY). | 3 |
| FIGUUR 2-1, CONTINUOUS CONTROL. | 7 |
| FIGUUR 2-2, CONTINUOUS SECURITY PYRAMID. | 8 |
| FIGUUR 2-3, VALUE CHAIN OF PORTER, BRON: [MICHAEL PORTER 1998]. | 9 |
| FIGUUR 2-4, RECURSIEVE VALUE CHAIN OF PORTER, BRON: [MICHAEL PORTER 1998]. | 10 |
| FIGUUR 2-5, RECURSIEVE VALUE CHAIN OF PORTER, BRON: [MICHAEL PORTER 1998]. | 11 |
| FIGUUR 2-6, DE OPBOUW VAN EEN VALUE SYSTEM. | 11 |
| FIGUUR 2-7, DE OPBOUW VAN EEN BUSINESS VALUE CHAIN. | 12 |
| FIGUUR 2-8, DE DRIE PERSPECTIEVEN VAN INFORMATION SECURITY. | 13 |
| FIGUUR 2-9, RISICOBEGRIPPEN. | 14 |
| FIGUUR 2-10, VALUE SYSTEM BEGRIPPEN. | 15 |
| FIGUUR 4-1, VERANDERPARADIGMA. | 21 |
| FIGUUR 4-2, VERANDERPARADIGMA - BEELDVORMING. | 22 |
| FIGUUR 4-3, VERANDERPARADIGMA- MACHTSVERHOUDING. | 24 |
| FIGUUR 4-4, VERANDERPARADIGMA - ORGANISATIE. | 27 |
| FIGUUR 4-5, VERANDERPARADIGMA - RESOURCES. | 29 |
| FIGUUR 5-1, CONTINUOUS SECURITY PYRAMID. | 35 |
| FIGUUR 5-2, CONTINUOUS SECURITY PYRAMID AFGEBEELD OP HET DEVOPS LEMNISCAAT. | 35 |
| FIGUUR 5-3, CONTINUOUS SECURITY PYRAMID MET DELIVERABLES EN TE BEANTWOORDEN VRAGEN. | 36 |
| FIGUUR 5-4, CONTINUOUS SECURITY PYRAMID MODEL AFGEBEELD OP CONTINUOUS CONTROL MODEL. | 37 |
| FIGUUR 5-5, QUALITY CONTROL & ASSURANCE MODEL. | 37 |
| FIGUUR 5-6, RECURSIVE VALUE CHAIN. | 38 |
| FIGUUR 5-7, INFORMATION SECURITY VALUE CHAIN. | 39 |
| FIGUUR 5-8, INFORMATION SECURITY VALUE SYSTEM. | 39 |
| FIGUUR 5-9, INFORMATION SECURITY PRACTICES. | 40 |
| FIGUUR 5-10, INFORMATION SECURITY VALUE SYSTEM OVERVIEW. | 41 |
| FIGUUR 5-11, SERVICE VALUE CHAIN. | 41 |
| FIGUUR 5-12, DEVELOPMENT VALUE CHAIN. | 42 |
| FIGUUR 5-13, CONTINUOUS SECURITY PYRAMID AFGEBEELD OP DE ISVS, DVS EN SVS MODELLEN. | 42 |
| FIGUUR 5-14, INFORMATION SECURITY PERSPECTIEVEN. | 43 |
| FIGUUR 6-1, CONTINUOUS SECURITY VALUE STREAM. | 45 |
| FIGUUR 6-2, USE CASE DIAGRAM VOOR CONTINUOUS SECURITY. | 47 |
| FIGUUR 7-1, INFORMATION SECURITY PRACTICES. | 53 |
| FIGUUR 7-2, INFORMATION SECURITY VALUE STREAMS. | 54 |
| FIGUUR 8-1, GOVERNANCE SECURITY PRACTICES. | 55 |
| FIGUUR 9-1, RISICO SECURITY PRACTICES. | 69 |
| FIGUUR 9-2, CRAMM MODEL. | 81 |
| FIGUUR 9-3, ASSET REGISTER. | 93 |
| FIGUUR 9-4, RISICOLEVENSCYCLUS. | 95 |
| FIGUUR 10-1, QUALITY SECURITY PRACTICES. | 109 |
| FIGUUR 10-2, MONITORARCHITECTUUR VOOR DE MONITORING VAN DE EFFECIVITEIT VAN CONTROLS. | 112 |
| FIGUUR 11-1, POSITIONERING AGILE EN AGILE SCRUM. | 125 |
| FIGUUR 11-2, AGILE SCRUM ONTWIKKELPROCES. | 130 |
| FIGUUR 11-3, AGILE SCRUM TEAMS. | 130 |
| FIGUUR 11-4, CONTINUOUS SECURITY USE CASE DIAGRAM. | 134 |
| FIGUUR 12-1, POSITIONERING DEVOPS. | 137 |
| FIGUUR 12-2, CONTINUOUS SECURITY AFGEBEELD OP HET DEVOPS LEMNISCAAT. | 138 |
| FIGUUR 12-3, CONTINUOUS SECURITY USE CASE DIAGRAM. | 141 |
| FIGUUR 13-1, DEVOPS CE-SPIDER MODEL. | 148 |

| | |
|---|-----|
| FIGUUR 13-2, DEVOPS CY-SPIDER MODEL. | 151 |
|---|-----|

Tabellen

| | |
|--|----|
| TABEL 1-1, CONTINUOUS EVERYTHING ASPECTEN. | 2 |
| TABEL 1-2, BIJLAGEN. | 5 |
| TABEL 3-1, VEEL VOORKOMENDE PROBLEMEN BIJ HET HANTEREN VAN CONTINUOUS SECURITY. | 18 |
| TABEL 6-1, TERMEN PER ISVS USE CASE. | 46 |
| TABEL 6-2, USE CASE TEMPLATE. | 48 |
| TABEL 6-3, USE CASE VOOR CONTINUOUS SECURITY. | 52 |
| TABEL 8-1, USE CASE 'VERKRIJG TOP MANAGEMENT COMMITMENT'. | 57 |
| TABEL 8-2, VOORBEELD VAN EEN STATEMENT OF COMMITMENT. | 58 |
| TABEL 8-3, BEPAAL INTERESTED PARTY. | 59 |
| TABEL 8-4, INTERESTED PARTIES - INVLOED EN BELANG. | 60 |
| TABEL 8-5, INTERESTED PARTIES - REGISTER. | 60 |
| TABEL 8-6, INTERESTED PARTIES – REGISTER UITLEG. | 61 |
| TABEL 8-7, SCOPE - USE CASE. | 62 |
| TABEL 8-8, SCOPE - DEFINITIE. | 63 |
| TABEL 8-9, SCOPE - USE CASE. | 64 |
| TABEL 8-10, DOELEN. | 65 |
| TABEL 8-11, INFORMATION SECURITY POLICY - USE CASE. | 67 |
| TABEL 8-12, CODE OF CONDUCT VOORBEELD VOOR BUSINESS EN IT. | 68 |
| TABEL 8-13, CODE OF CONDUCT VOORBEELD VOOR IT. | 68 |
| TABEL 9-1, USE CASE 'BEPAAL INTERNAL ISSUES'. | 71 |
| TABEL 9-2, VOORBEELDEN VAN INTERNAL ISSUES FACTOREN. | 72 |
| TABEL 9-3, VOORBEELDEN VAN IPOPS FACTOREN. | 73 |
| TABEL 9-4, VOORBEELD IPOPS CLASSIFICATIE TEMPLATE. | 73 |
| TABEL 9-5, USE CASE 'BEPAAL EXTERNAL ISSUE'. | 76 |
| TABEL 9-6, VOORBEELDEN VAN EXTERNAL ISSUES. | 76 |
| TABEL 9-7, VOORBEELDEN VAN PESTLE FACTOREN. | 77 |
| TABEL 9-8, VOORBEELD PESTLE CLASSIFICATIE TEMPLATE. | 77 |
| TABEL 9-9, USE CASE 'BEPAAL CRAMM ISSUE'. | 79 |
| TABEL 9-10, VOORBEELDEN VAN CRAMM BEDREIGINGEN. | 81 |
| TABEL 9-11, TEMPLATE CRAMM-ANALYSE. | 82 |
| TABEL 9-12, USE CASE 'BEPAAL RISK CRITERIA'. | 84 |
| TABEL 9-13, RISICO PRIORITEIT CRITERIA. | 86 |
| TABEL 9-14, RISICO PRIORITEIT CRITERIA. | 86 |
| TABEL 9-15, RISICO PRIORITEIT CRITERIA. | 87 |
| TABEL 9-16, INTERNE AUDIT FINDING CRITERIA. | 87 |
| TABEL 9-17, DE IMPACTCODE VAN INFORMATIE SECURITY INCIDENTEN. | 88 |
| TABEL 9-18, DE IMPACTCODE VAN INFORMATIE SECURITY INCIDENTEN. | 90 |
| TABEL 9-19, INFORMATION SECURITY INCIDENT PRIORITEITENTABEL. | 90 |
| TABEL 9-20, INFORMATION SECURITY INCIDENT EVIDENCE MATRIX. | 91 |
| TABEL 9-21, USE CASE 'BEPAAL INFORMATION SECURITY ASSETS'. | 92 |
| TABEL 9-22, USE CASE 'IDENTIFICEER RISICO'S'. | 94 |
| TABEL 9-23, GEÏDENTIFICEERD RISICO. | 95 |
| TABEL 9-24, TEMPLATE RISICO IDENTIFICATIE. | 96 |
| TABEL 9-25, UITLEG VAN RISK IDENTIFICATION ITEMS. | 96 |
| TABEL 9-26, USE CASE 'IDENTIFICEER RISICO'S'. | 97 |
| TABEL 9-27, TEMPLATE RISK IDENTIFICATION. | 98 |
| TABEL 9-28, TEMPLATE RISK ASSESSMENT. | 98 |
| TABEL 9-29, TEMPLATE RISK ASSESSMENT. | 98 |

| | |
|---|-----|
| TABEL 9-30, USE CASE 'RISK TREATMENT OPTIONS RISICO'S' | 100 |
| TABEL 9-31, MASR TREATMENT OPTIONS | 100 |
| TABEL 9-32, UITLEG TREATMENT OPTIONS | 101 |
| TABEL 9-33, USE CASE 'RISICO CONTROLS TOEKENNEN AAN RISICO'S' | 102 |
| TABEL 9-34, MASR TREATMENT OPTIONS | 103 |
| TABEL 9-35, USE CASE 'RISICO CONTROLS TOEKENNEN AAN RISICO'S' | 105 |
| TABEL 9-36, USE CASE 'OPSTELLEN RISK TREATMENT PLAN VOOR EEN CONTROL' | 107 |
| TABEL 9-37, TREATMENT PLAN TEMPLATE | 108 |
| TABEL 10-1, USE CASE 'MONITOR EFFECTIVITEIT CONTROLS' | 111 |
| TABEL 10-2, USE CASE 'INTERNE AUDITPLANNING' | 114 |
| TABEL 10-3, TEMPLATE INTERNE AUDIT | 115 |
| TABEL 10-4, USE CASE 'INTERNE AUDITCRITERIA' | 116 |
| TABEL 10-5, USE CASE 'UITVOERING INTERNE AUDIT' | 118 |
| TABEL 10-6, USE CASE 'RAPPORTAGE INTERNE AUDIT' | 119 |
| TABEL 10-7, USE CASE 'INFORMATION SECURITY INCIDENTEN' | 121 |
| TABEL 10-8, USE CASE 'INFORMATION SECURITY NC' | 122 |
| TABEL 10-9, USE CASE 'INFORMATION SECURITY NC' | 124 |
| TABEL 10-10, CSI REGISTER VOORBEELD | 124 |
| TABEL 11-1, EFFECTIVITEIT ASPECTEN VAN HET AGILE SYSTEEMONTWIKKELING | 127 |
| TABEL 11-2, EFFICIËNTIE ASPECTEN VAN HET AGILE SYSTEEMONTWIKKELING | 128 |
| TABEL 11-3, AFBEELDING CONTINUOUS SECURITY OP AGILE SCRUM | 134 |
| TABEL 11-4, AFBEELDING CONTINUOUS SECURITY OP AGILE SCRUM ARTEFACTS | 135 |
| TABEL 11-5, AFBEELDING CONTINUOUS SECURITY OP AGILE SCRUM EVENTS | 136 |
| TABEL 12-1, CONTINUOUS EVERYTHING ASPECTEN | 138 |
| TABEL 12-2, AFBEELDING CONTINUOUS SECURITY OP DEVOPS | 142 |
| TABEL 13-1, DEVOPS CE-MODEL | 145 |
| TABEL 13-2, CONTINUOUS EVERYTHING | 146 |
| TABEL 13-3, CMMI LEVELS VOOR CONTINUOUS EVERYTHING | 147 |
| TABEL 13-4, PR-ORG-009. VOLWASSENHEIDSNIVEAUS | 148 |
| TABEL 13-5, CY MATURITY CHARACTERISTICS | 151 |

Bijlagen

| | |
|----------------------------|-----|
| BIJLAGE A, LITERATUURLIJST | 155 |
| BIJLAGE B, BEGRIPPENLIJST | 159 |
| BIJLAGE C, AFKORTINGEN | 175 |
| BIJLAGE D, WEBSITES | 179 |
| BIJLAGE E, INDEX | 181 |

Voorwoord

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

Veel van mijn ervaringen heb ik ook al gedeeld in de artikelen op www.ITpedia.nl. 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 |
| • H. (Hans) Hamhuis | Argis IT Consultants |
| • 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 security binnen uw eigen organisatie.

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

1 Introductie

Leeswijzer:

Dit hoofdstuk beschrijft de achtergrond van dit boek (1.1) de beoogde doelgroep (1.2), de structuur (1.3) en tenslotte enkele tips voor het hanteren van dit boek (1.4) en de leeswijzer (1.5).

1.1 Doel

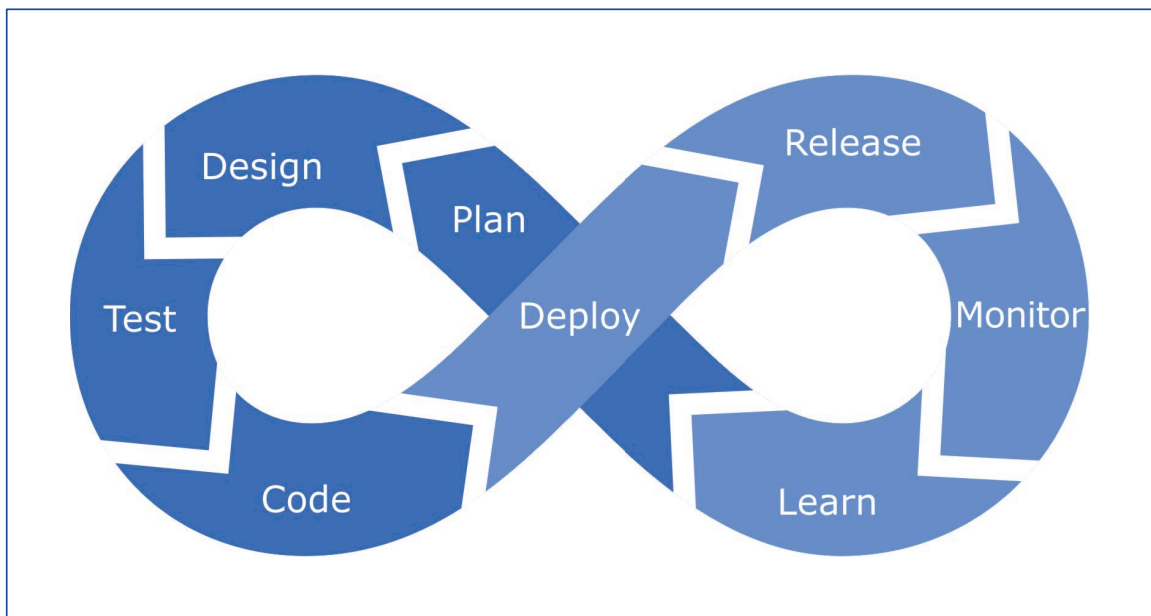
De doelstelling van dit boek is het geven van basiskennis ten aanzien van continuous security en tips en trucs voor het toepassen van dit aspectgebied van continuous everything.

1.2 Doelgroep

De doelgroep van dit boek zijn alle betrokken functies bij de DevOps teams. Dit omvat zowel de auditors, kwaliteitsmedewerkers, 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.

1.3 Achtergrond

Dit boek bevat verschillende technieken om op een continue wijze invulling te geven aan compliancy zodat deze kwaliteit van het informatiesysteem meegroeit tijdens de voortbrenging. Continuous security is een integraal onderdeel van de DevOps Lemniscaat zoals weergegeven in [Figuur 1-1](#) en wel in alle stappen die doorlopen worden.



Figuur 1-1, DevOps Lemniscaat.

De DevOps lemniscaat geeft een overzicht van de te doorlopen fasen om software continu voort te brengen. De DevOps Lemniscaat is daardoor een goede basis voor het definiëren van het concept van continuous security. Continuous security is niet één geïsoleerd aspectgebied van het Continuous Everything (CE) concept maar heeft invloed op alle stappen in het DevOps lemniscaat.

Het CE-concept beschrijft alle fasen van de DevOps Lemniscaat in de vorm van continu 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 Security (-) |
| 5 | Continuous Deployment (Deploy) | 10 | Continuous Assessment (-) |

Tabel 1-1, Continuous everything aspecten.

Continuous auditing (9) en continuous assessment (10) zijn niet expliciet in de DevOps Lemniscaat weergegeven, net als andere continuous aspectgebieden zoals continuous documentation en continuous robotics om wille van de eenvoud van de DevOps Lemniscaat.

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

Continuous security is een Agile wijze van het realiseren van de vereiste controls tijdens het vormgeven van een informatiesysteem door op het juiste tijdstip het gedrag, de functionaliteit en de kwaliteit van het informatiesysteem te definiëren die de onderkende information security risico's mitigeren of elimineren. Alle aspecten van de DevOps Lemniscaat hebben direct of indirect een relatie met continuous security omdat deze holistisch wordt vormgegeven. Dit wil zeggen dat continuous security betrekking heeft op zowel het informatiesysteem (Technology) als het voortbrengingsproces (Process) als de kennis en de kunde (People). Daarmee geeft continuous security dus een design (ontwerp) op PPT niveau.

Een belangrijke basis voor het borgen van de controls van een informatiesysteem is het design (ontwerp) ervan. Het bestaansrecht van een design voor een informatiesysteem is de laatste jaren bij veel organisaties echter in twijfel getrokken. De klassieke rechtvaardiging van het bundelen van informatie over een informatiesysteem en het betrekken van alle stakeholders wordt gezien als achterhaald door de Agile manier van werken en het idee van 'the three amigo development strategy'. Deze strategie houdt in dat vanuit drie vakdisciplines: business, development en testen van tevoren gekeken wordt naar een increment dat moet worden gebouwd. Daarmee wordt de 'hoe' en de 'wat' vraag beter uitgewerkt en kan er consensus bereikt worden over de Definition of Done (DoD) van het increment. Hierbij wordt echter voorbijgegaan aan de andere klassieke rechtvaardiging van een ontwerp en dat is dat een ontwerp ook bedoeld is voor de control-functie die voorkomt dat information security risico's materialiseren door gebrek aan tegenmaatregelen. Deze risico's betreffen verschillende aspectgebieden waaronder het niet nakomen van wet- en regelgeving verplichtingen. Een ander belangrijk aspectgebied van controls zijn die van information security zoals beveiliging, integriteit en vertrouwelijkheid van informatie.

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

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

System of Records

De SoR zijn informatiesystemen van de backoffice die de financiën, logistiek, voorraad en Human Resource Management (HRM) taken vervullen.

Deze systemen moet voldoen aan de eisen van information security in termen van Confidentiality, Integrity en Availability (CIA). Dit betekent onder andere dat er designs vereist worden die aangeven hoe de financiële gegevens worden voortgebracht en wat de interfaces zijn van de verschillende betrokken informatiesystemen. Het zijn in het algemeen informatiesystemen die onderdeel zijn van een keten van informatiesystemen. Deze systemen vereisen een van tevoren goed overwogen aanpak en dus een design waarin information security een belangrijke rol speelt.

System of Engagement

De SoE informatiesystemen zijn gericht op verkoopkanalen naar consumenten waarbij vooral gedacht moet worden aan webshops en apps voor de smartphones. Deze applicaties zijn eenvoudig te voorzien van een nieuwe release, versie en patch. Deze informatiesystemen zijn meestal geen integraal onderdeel van een keten maar veeleer de eindpunten van een keten. Dit zijn vaak ook de voorbeelden uit de publicaties over Agile en Development & Operations (DevOps). Voor deze informatiesystemen is het duidelijk dat een van tevoren doordacht design (upfront design) minder hard nodig is en vaak volstaan kan worden met een groeiend design (emerging design). Wel is het van belang deze eindpunten te beschouwen als mogelijke lekken in de information security. Daarom is het voor deze SoE informatiesystemen handig om meer dan alleen een verzameling user stories te hebben. Information security moet ook voor deze informatiesystemen een integraal onderdeel vormen. Tevens vormen de losse user stories geen toegankelijke beschrijving van de information security van een informatiesysteem. Er is dus nog steeds behoefte aan een design dat overzicht en inzicht geeft van de functionaliteit, kwaliteit en de werking van het informatiesysteem alsmede de information security. Vooral de userinterface en interfaces met databronnen vormen hierbij belangrijke aspectgebieden van beveiliging en daarmee continuous security.

System of Intelligence

Daarnaast zijn er nog de Business Intelligence (BI) oplossingen. Dit zijn de rapportages, data-analyse tools en dergelijke. Voor deze toepassingen geldt hetzelfde als de SoE informatiesystemen. Ze zijn de weergave van informatie uit de SoR en zijn makkelijker aan te passen. Evenwel zijn het lekken van data en inbraak ook risico's. Afhankelijk van de waarde van de informatie die toegankelijk wordt gesteld is het risico hoger of lager. Van informatie moet dus ook bekend zijn wat de risico's zijn en welke tegenmaatregelen getroffen moeten worden.



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

Noodzaak continuous security

Voor alle drie de typen (SoR, SoE en SoI) informatiesystemen is er dus een behoefte aan een bepaalde mate van control en daarmee aan continuous security.

Er is meer dan alleen een set van user stories nodig om overzicht en inzicht te krijgen en te houden van de control-requirements die aan het informatiesysteem worden gesteld.

Anders kunnen de risico's en de impact van aanpassing en uitbreiding van het informatiesysteem niet tijdig en correct worden onderkend. Evenwel moet voorkomen worden dat de implementatie van de tegenmaatregelen (controls) van de risico's de Agility van het voortbrengingsproces te niet doet. Dit betekent dat controls niet alleen incrementeel en iteratief moeten worden vormgegeven en bewaakt (continuous security) maar dat de controls ook geselecteerd moeten worden op basis van het gewogen risico, dus niet te veel maar zeker ook niet te weinig. De mate waarin het design van de control gedefinieerd moet zijn, verschuift van veel bij SoR naar minder bij SoE en nagenoeg niets bij SoI. Maar ook bij een SoR kan het design van controls worden opgedeeld in lagen die meer upfront (van tevoren) en emerging (in sprints) gedefinieerd worden.

1.4 Structuur

Dit boek bespreekt hoe continuous security vorm te geven aan de hand van het Information Security Value System (ISVS) model. Voordat dit model wordt besproken wordt eerst invulling gegeven aan de definities, de verankering en architectuur van continuous security. Daarna volgt de bespreking van dit model.

1.4.1 Hoofdstuk 2: Basisconcepten en basisbegrippen

Dit hoofdstuk bespreekt de basisconcepten en de basisbegrippen.

1.4.2 Hoofdstuk 3: Continuous security definitie

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

1.4.3 Hoofdstuk 4: Continuous security verankering

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

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

1.4.4 Hoofdstuk 5: Continuous security architectuur

Dit hoofdstuk beschrijft de architectuurprincipes en – modellen voor continuous security. De architectuurmodellen betreffen de volgende modellen:

- ISVS model (ISO 27001)
- Service Value System (SVS) model (ITIL 4)
- Development Value System (DVS) model (Agile Scrum)
- DevOps Lemniscate model (DevOps procesmodel)

1.4.5 Hoofdstuk 6: Continuous security ontwerp

Het ISVS bestaat uit een information security value chain die invulling geeft aan de information security value streams. De value streams worden op basis van use cases beschreven. De relaties tussen de use cases wordt weergegeven in een use case diagram.

1.4.6 Hoofdstuk 7: Continuous security best practices

Dit hoofdstuk beschrijft hoe het ISVS te construeren en bespreekt een aantal continuous security best practices.

1.4.7 Hoofdstuk 8: Governance security practices

Dit hoofdstuk bespreekt de governance security practices. Dit zijn de best practices die de ISVS value chain besturen. Het betreft de practices voor het verkrijgen van top management commitment, bepalen van interested parties, bepalen van de scope, bepalen van doelen en bepalen van de information security policy. Voor elke security practice wordt de use case besproken, alsmede de definitie, de doelstelling en waarvan toepassing een voorbeeld.

1.4.8 Hoofdstuk 9: Risico security practices

De risico security practices bestaan uit de operationele best practices van het ISVS. Het betreft de practices voor het bepalen van de issues, risicocriteria en information assets, het identificeren van risico's, het verrichten van een risk assessment en risk treatment alsmede het realiseren van controls. Voor elke security practice wordt de use case besproken, alsmede de definitie, de doelstelling en waarvan toepassing een voorbeeld.

1.4.9 Hoofdstuk 10: Quality security practices

De quality security practices bestaan uit de interne audit en het continue verbeteren best practices van het ISVS. Voor elke security practice wordt de use case besproken, alsmede de definitie, de doelstelling en waarvan toepassing een voorbeeld.

1.4.10 Hoofdstuk 11: Continuous security versus Agile Scrum

Het in isolatie benaderen van continuous security is zinloos. Er zijn te veel koppelvlakken met systeemontwikkeling (Agile Scrum) zodat de integratie van de vakgebieden noodzakelijk is. Dit hoofdstuk beschrijft de essentie van deze koppeling.

1.4.11 Hoofdstuk 12: Continuous security versus DevOps

De koppeling tussen information security en Agile Scrum is een hele belangrijke vooruitgang. Nochtans is er ook een koppeling met operations nodig om continuous security te operationaliseren. Dit hoofdstuk bespreekt hoe continuous security te integreren met DevOps aan de hand van het concept van continuous everything.

1.4.12 Hoofdstuk 13: Continuous security assessment

De volwassenheid van continuous security is in dit hoofdstuk meetbaar gemaakt aan de hand van een continuous security assessment.

1.5 Bijlagen

De bijlagen bevatten belangrijke informatie die helpt bij het beter begrijpen van continuous security.

| 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 is 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 informatiemangement", 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. |
| Oirsouw 2001 | R.R. van Oirsouw, J. Spaanderman, C. van Arendonk, "Informatiseringseconomie", ISBN 90 395 1393 7, 2001. |

| Referenties | Publicaties |
|---------------|--|
| 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 existence 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 |
| SBB | System Building Block |

| Afkorting | Betekenis |
|-----------|--|
| 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 · 10, 56, 59, 61, 64, 66, 70, 74, 75, 78, 79, 83, 91, 94, 97, 99, 102, 104, 106, 110, 113, 116, 117, 119, 120, 122, 123, 175

5

5-times why · 18

A

A/B testing · 146, 159
 acceptatiecriteria · 45, 51, 83, 84, 86, 116, 131, 139
 acceptatiecriterium · 160, 164
 acceptatietest · 159
 accountable · 56, 57
 actor · 48, 50, 56, 59, 61, 62, 64, 66, 70, 75, 79, 83, 91, 92, 94, 97, 99, 102, 104, 105, 106, 110, 113, 116, 117, 119, 120, 122, 123
 adaptive software development · 128
 affinity · 159
 AFM · 175
 Agile · 25, 147, 159, 171, 172
 - coach · 1
 - infrastructure · 159
 - manifesto · 125
 - methode · 128
 - modeling · 128
 - proces · 147
 - release train · 25
 - unified process · 128
 Agile Scrum · 24, 27, 163
 Agile Scrum framework · 24, 25
 Agile Scrum proces · 24
 Algemene Verordening Gegevensbescherming · Zie AVG
 alternate path · 159
 Amazon Web Services · Zie AWS
 ambitieniveau · 8, 18, 19, 135, 136
 Andon cord · 159
 anker · 21
 annotatie · 146
 anomaly detection technique · 159
 antifragility · 160
 anti-pattern · 22, 23, 26, 28, 29, 160
 apenrots · 21
 applicatie · 18, 23, 24, 146
 applicatiebeheer · 166
 applicatiecomponent · 167
 architectuur · 127
 architectuurprincipe · 31

artefact · 46, 160, 162
 artefact repository · 160
 assessment · 19, 170
 assessment methode · 72, 76, 81
 asset · 4, 45, 49, 51, 61, 63, 67, 68, 69, 72, 79, 80, 82, 85, 91, 92, 94, 111, 112, 115, 116, 149, 150
 - categorie · 45, 91, 92
 - groep · 45, 79, 80, 85, 92, 94, 97, 104, 116
 - owner · 132
 - register · 45, 63, 93, 96
 - type · 64, 85, 96
 audit · 117
 - basis · 116
 - criteria · 109, 115, 116, 117
 - log · 118
 - plan · 109, 113
 auditing · 146
 automated test · 160
 Autoriteit Financiële Markten · Zie AFM
 availability · 3, 14, 50, 51, 55, 64, 65, 68, 70, 88, 89, 90, 161
 AVG · 14, 76, 77, 175
 awareness · 38, 49, 51, 58, 66, 67, 72
 AWS · 175

B

backlog · 129
 backlog item · 169
 bad apple theory · 160
 bad path · 160
 Balanced Score Card · 10, Zie BSC
 barrier · 61
 baseline · 133, 146
 BDD · 146, 160, 175
 bedreiging · 13, 14, 69, 72, 75, 76, 79, 80, 81, 82, 85
 bedrijfsproces · 127, 146
 beeldvorming · 4, 21, 22, 23, 27, 28
 Behavior Driven Development · Zie BDD
 belastingwet · 22
 beschikbaarheid · 34
 best practice · 21, 148, 161
 betrouwbaarheid · 127
 beveiliging · 147
 bevindingen · 49, 52, 83, 85, 87, 109, 114, 115, 118, 120
 bevoegdheden · 4, 21
 bewijsmateriaal · 9, 18, 85, 87
 BI · 175
 BI-model · 2
 binary · 160
 BI-oplossing · 3
 blameless post mortem · 146, 160
 blamelessness · 160
 blind pokersessie · 74
 blue/green deployment · 160

blue/green environment · 146
 Body of Knowledge · Zie BOK
 BOK · 175
 bottom-up · 21
 branching · 146, 161
 broken build · 146, 160
 brown field · 160
 BSC · 8, 175
 build · 146, 160, 161, 162, 171
 built-in failure mode · 146
 business

- devOps · 146
- functie · 131
- value · 161, 163
- value chain · 11, 12, 15, 38, 40, 55, 61, 62, 63
- value stream · 26

 Business Intelligence · Zie BI
 Business Value System · Zie BVS
 BVS · 12, 13, 33, 38, 175

C

CA · 175
 CAB · 175
 CAMS · 162, 175
 canary releasing · 146, 161
 capability · 162
 Capability Maturity Model Integration · Zie CMMI
 capaciteit · 132, 161
 CCTA Risk Assessment Method
 Methodology · Zie CRAMM
 CD · 146, 148, 161, 165, 175
 CE · 148, 175
 CEM · 175
 CEMLI · 175
 CE-model · 145
 Central Event Monitor · Zie CEM
 Central Processing Unit · Zie CPU
 CEO · 57, 58, 175
 CFO · 57, 58, 175
 Change Advisory Board · Zie CAB
 change category · 161
 change schedule · 161
 check-in · 146
 checkpunt · 151
 Chief Executive Officer · Zie CEO
 Chief Finance Officer · Zie CFO
 Chief Information Officer · Zie CIO
 Chief Technology Officer · Zie CTO
 chores · 131
 CI · 145, 146, 148, 161, 165, 175
 CI/CD secure pipeline · 2, 12, 24, 111, 112, 140, 142, 143, 149, 150
 CIA · 14, 45, 46, 50, 63, 64, 65, 68, 69, 70, 71, 72, 75, 79, 82, 85, 89, 90, 94, 96, 97, 98, 101, 103, 104, 105, 106, 108, 111, 112, 116, 120, 125, 135, 136, 137, 175
 CIA-aspect · 64
 CIA-doelstelling · 64
 CIA-matrix · 14, 63, 65, 68, 69, 79, 82, 94, 96, 101, 104, 105, 106, 108, 111, 112, 116, 125, 137
 CIO · 58, 175
 CISO · 57
 CL · 146, 175
 classificatie matrix · 74, 78, 82
 cloud · 161
 cloud configuration file · 161
 cloud service · 161
 cluster immune system release pattern · 161
 CM · 146, 149, 151, 175
 CMDB · 175
 CMMI · 145, 147, 148, 149, 175
 CMS · 175
 CN · 175
 CO · 146, 175
 CoC · 45, 51, 66, 67, 68, 72, 91, 175
 code branch · 161
 Code of Conduct · Zie CoC
 code review form · 161
 codereview · 131
 codified NFR · 161
 collaboration · 161
 commit code · 161
 commit stage · 161
 commitment statement · 45, 56, 57
 Communities of Practice · Zie CoP
 competence · 159, 164
 Competitive Advantage · Zie CA
 Competitive Response · Zie CR
 Completeness / Accurateness · Zie %C/A
 compliance · 146, 162
 compliance checking · 162
 compliancy · 1, 32, 33, 38, 147, 162
 compliancy officer · 162
 component · 165, 168, 171
 component uitval · 65
 confidentiality · 3, 14, 50, 51, 55, 64, 65, 68, 70, 88, 89, 90, 175
 Confidentiality, Integrity & Availability · Zie CIA
 configuratiebeheer · 146
 Configuration Item · Zie CI
 configuration management · 162
 Configuration Management DataBase · Zie CMDB
 Configuration Management System · Zie CMS
 Configuration, Extention, Modification, Localisation, Integration · Zie CEMLI
 container · 162
 context · 24, 145
 contingentiefactor · 8, 48, 127
 continuity · 161, 167
 continuous

- assessment · 2, 138, 142
- auditing · 138, 142
- control model · 7, 13, 31, 36, 37
- delivery · 148
- deployment · 2, 138, 142
- design · 2, 138, 142

- documentation · 2
- everything · 1, 17, 145, 146, 147, 148
- improvement · 12
- integration · 2, 128, 138, 142
- learning · 2, 138, 142
- monitoring · 2, 138, 142
- planning · 2, 138, 142
- security · 1, 2, 5, 17, 22, 23, 24, 31, 34, 46, 47, 48
- security pyramid · 12, 13, 21, 22, 24, 25, 31, 34, 35, 36
- security roadmap · 25
- sssessment · 2
- testing · 2, 138, 142

Continuous

- Improvement · 172
- Learning · 172

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

contract · 126, 132

control · 2, 148, 162, 170

- backlog · 15
- evidence database · 46, 112
- functie · 2
- label · 72, 86
- model · 36

Conway's law · 162

cookbook · 74, 78, 82, 96, 98, 101, 104, 108, 112, 115

CoP · 25, 175

counter measure · 166

CP · 175

CPU · 175

CR · 175

CRAMM · 175

CRAMM bedreiging · 81

CRAMM issue · 79

CRAMM issue register · 45, 94, 96

CRC · 175

CRC code · 65

CS · 175

CSI register · 46, 49, 123, 124

CT · 146, 175

CTO · 175

cultural debt · 162

Culture, Automation, Measurement and Sharing · Zie CAMS

CY · 175

cycle time · 146, 162

Cyclic Redundancy Check · Zie CRC

D

data · 148

data transport · 65

dataverlies · 65

debt · 162

declarative programming · 162

defect · 147, 167

defect tracking · 163

Definition of Done · Zie DoD

Definition of Ready · Zie DoR

Definitional Uncertainty · Zie DU

Definitive Media Library · Zie DML

deliverable · 36

Deming · 127

Demming wheel · 166

deployment · 159

deployment pipeline · 161

deployment team · 131

design · 2, 162, 173

Dev engineer · 1

development · 159, 160, 163, 165, 166, 168, 170, 171, 172, 173

Development & Operations · Zie DevOps

development ritual · 163

development team · 24, 25, 129, 130, 131

Development Value System · Zie DVS

Development, Test, Acceptance and Production · Zie DTAP

DevOps · 1, 2, 3, 5, 11, 21, 24, 25, 26, 29, 35, 138, 145, 147, 148, 151, 159, 161, 163, 169, 173, 175

- engineer · 26, 159, 161, 162, 163, 170, 171, 172
- Lemniscaat · 1, 2, 35, 138
- team · 1, 24, 25, 26, 159, 160, 161, 162, 164, 168, 173
- werkwijze · 1
- WoW · 24, 25

digitale hacker · 73

Digitalisering · 76

DML · 176

DNS · 176

DoD · 2, 49, 131, 136, 147, 149, 165, 176

Domain Name System · Zie DNS

doorzettingsvermogen · 127

DoR · 147, 176

downward spiral · 163

DTAP · 170, 176

DTAP environments · 170

DU · 176

duurzaamheid · 76, 77

DVS · 4, 7, 10, 11, 12, 33, 36, 38, 176

E

E2E · 147, 176

effectief · 24

efficiënt · 24

eigenaarschap · 21, 24, 26

eindgebruiker · 131

eindresultaat · 131
 elimineren · 2, 8, 17, 28, 65, 68, 99, 128, 140
 e-mail pass around · 146, 163
 encryptie mechanisme · 61
 End User eXperience Monitoring · Zie EUX
 End-to-End · Zie E2E
 Enterprise Resource Planning · Zie ERP
 Enterprise Service Buss · Zie ESB
 Entity Relation Diagram · Zie ERD
 epic · 131, 147, 176
 Epic Solution Approach · Zie ESA
 epic user story · 131, 132
 ERD · 176
 ERP · 176
 error path · 163
 ESA · 176
 ESB · 176
 E-shaped · 146, 164
 ETL · 176
 EUX · 176
 event · 146, 168
 eventcatalogus · 131
 eventlogboek · 45
 evidence · 9, 13, 18, 22, 23, 26, 27, 28, 33, 34, 36, 38, 42, 45, 46, 57, 69, 83, 87, 90, 91, 110, 111, 112, 113, 117, 118, 133, 134, 142, 149, 150
 evidence collector · 46, 111, 112, 113, 149
 evidence criteria · 45, 83
 evolutionary project management · 128
 eXtensible Markup Language · Zie XML
 external issue · 45, 58, 59, 69, 75, 76, 78, 85, 94
 Extract Transform & Load · Zie ETL
 eXtreme Programming · Zie XP

F

failure · 160
 fast feedback · 21
 FAT · 131, 159, 176
 feature · 131, 147, 163, 164, 173
 feature driven development · 128
 feature request · 129
 Feature Solution Approach · Zie FSA
 feature toggle · 146, 163
 feedback · 146, 147, 162, 163, 166, 169, 172
 feedforward · 163
 financiële verslaglegging · 17
 finding criteria · 45, 83, 85, 87
 five times why methode · 18
 flexibiliteit · 129
 flow · 146, 147, 162, 165, 166, 167, 168, 169, 170, 172, 173
 focus · 128, 132
 framework · 168
 front-end applicatie · 24
 FSA · 176
 functie · 1, 25
 functiepunt · 133

functiepuntenanalysemethode · 133
 functionaliteit · 127, 128, 129
 Functionele AcceptatieTest · Zie FAT

G

Galbraith · 25
 GAT · 176
 Gaussian distribution · 159, 163
 GCC · 28, 176
 GDPR · 8, 22, 176
 gebruiker · 127, 128, 129, 133
 Gebruiker AcceptatieTest · Zie GAT
 gebruikersinterface · 48
 gebruikersorganisatie · 126, 127, 128
 Gene Kim · 163, 168, 172
 General Computer Controls · Zie GCC
 General Data Protection Regulation · Zie GDPR
 Generieke & Specifieke Acceptatiecriteria · Zie GSA
 gereedschap · 146
 gewenste gedrag · 17
 Gherkin language · 33, 111
 GIT · 176
 Given When Then · 164, Zie GWT
 Global Information Tracker · Zie GIT
 governance · 4, 11, 12, 25, 26, 27, 33, 40, 49, 53, 55, 69, 109
 Graphical User Interface · Zie GUI
 green build · 146
 green field · 164
 GSA · 176
 GUI · 176
 guild · 25
 GWT · 164, 176

H

Hand-off Readiness Review · Zie HRR
 happy flow · 45
 happy path · 159, 160, 164
 hardware · 132, 162, 165, 173
 Holocracy · 164
 horizontal splitting of feature · 164, 173
 HRM · 2, 29, 31, 132, 176
 HRM manager · 29
 HRR · 146, 176
 Human Resource Management · Zie HRM
 hypothesis driven development · 146

I

IaC · 159, 165, 176
 ICT · 165, 176
 ID · 176
 ideal test pyramid · 146, 172
 idempotent · 164
 Identifier · Zie ID

- impact · 4
 - impact rate · 74, 78
 - impact severity · 73, 74, 77, 78
 - imparative programming · 164
 - impediment · 131
 - incident · 129
 - incident criteria · 45, 83
 - incrementeel · 2, 4, 33
 - Independent, Negotiable, Valuable, Estimatable, Small and Testable · Zie INVEST
 - indringer · 65
 - informatiedrager · 49
 - informatiesysteem · 2, 3, 18, 23, 24, 28, 126, 127, 128, 132
 - Information assets, People, Organisation, Products and services, Systems and processes · Zie IPOPS
 - Information Communication Technology · Zie ICT
 - information radiator · 165
 - information security · 2, 3, 18
 - auditing engine · 46, 112, 150
 - awareness training · 66, 67
 - normenkader · 23
 - policy · 4, 45, 46, 49, 55, 56, 57, 64, 65, 66, 67, 72, 87, 97, 114, 118
 - value chain · 4, 39, 40
 - value stream · 19
 - 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 · 165
 - Infrastructure as Code · Zie IaC
 - infrastructure component · 165
 - infrastructure management · 165
 - Infrastructure Risk · Zie IR
 - innovatie · 129
 - integriteit · 2
 - integrity · 3, 14, 50, 51, 55, 64, 65, 68, 70, 88, 89, 90, 175
 - interested parties register · 45, 63
 - interested party · 4, 13, 14, 40, 45, 49, 58, 59, 60, 61, 62, 63, 66, 85, 108, 109, 111, 112, 116, 125, 133, 137
 - internal issue · 70
 - International Standard On Assurance Engagements · Zie ISAE
 - interne audit · 5, 46, 49, 52, 85, 87, 109, 113, 114, 115, 116, 117, 118, 119, 123, 124
 - actielijst · 115
 - approach · 114, 117
 - doel · 114, 116
 - finding · 115
 - partijen · 114, 116
 - plan · 113, 114, 115, 116
 - rapport · 115
 - resultaat · 114, 117
 - scope · 114, 117
 - tijd · 114, 117
 - INVEST · 164, 176
 - IP address · 165
 - IPOPS · 45, 69, 70, 71, 72, 73, 74, 176
 - IR · 176
 - ISAE 3402 · 8
 - I-shaped · 146, 164
 - ISMS · 176
 - ISO · 176
 - ISO 27001 · 4, 7, 8, 10, 13, 23, 28, 38, 39, 42, 46, 56, 58, 61, 62, 67, 68, 72, 76, 87, 95, 101, 103, 104, 109, 111, 114, 115, 116, 117, 118, 122, 136, 142
 - issue logboek · 45, 50, 70, 74, 75, 78
 - issue type · 74, 78
 - IST · 166
 - IST – SOLL – Migratiepad · 7
 - ISVS · 4, 10, 12, 32, 33, 34, 35, 36, 176
 - IT · 163, 167, 172, 176
 - iteratief · 2, 4, 33
 - ITIL 4 · 10, 11, 176
 - ITSM · 167, 176
-
- ## J
- Java Virtual Machine · Zie JVM
 - Ji-Kotei-Kanketsu · Zie, Zie JKK
 - JIT · 165, 166, 176
 - JKK · 165, 176
 - JSON formaat · 27, 111
 - Just In Time · Zie JIT
 - JVM · 176
-
- ## K
- Kaizen · 165, 167
 - Kaizen Blitz (or Improvement Blitz) · 166
 - Kaizen in advance · 166
 - Kanban · 166, 167, 173
 - Ken Schwaber · 24
 - keten · 24, 146
 - Key Performance Indicator · Zie KPI
 - kibana dashboard · 166
 - KPI · 55, 64, 150, 166, 167, 173, 176
 - Kritieke Succes Factor · Zie KSF
 - KSF · 45, 55, 64, 65, 68, 70, 176
 - kwaliteit · 25, 127
 - kwaliteits
 - aspect · 32
 - borging · 33
 - criteria · 147
 - eis · 132

L

lag monitor · 7
 lagenstructuur · 36
 LAN · 177
 latent defect · 166
 Launch Readiness Review · Zie LRR
 launching guidance · 166
 LCM · 177
 LDAP · 177
 lead en lag rapportage · 110
 Lead Time · 166, Zie LT
 Lean · 172, 173
 Lean indicator · 10
 Lean software development · 128
 Lean tool · 166
 learning culture · 167
 levenscyclus · 31, 32, 34, 146
 leverancier · 132
 lifecycle · 163, 165
 LifeCycle Management · Zie LCM
 Lightweight Directory Access Protocol · Zie LDAP
 lijnmanager · 1
 lijnorganisatie · 132
 Local Area Network · Zie LAN
 log · 146, 168
 logging level · 167
 loosely coupled architecture · 167
 loosely coupled services · 167
 LRR · 146, 166, 177
 LT · 10, 56, 59, 61, 64, 66, 70, 74, 75, 78, 79, 83, 91, 94, 97, 99, 102, 104, 106, 110, 113, 116, 117, 119, 120, 122, 123, 166, 177

M

machtsverhouding · 4, 21, 23, 24, 26, 27, 28
 managed object · 46
 management · 148
 Management Information · Zie MI
 manifesto · 125, 128
 manual testing · 149
 manufacturing process · 173
 marker · 49
 MASR · 46, 69, 100, 103, 135, 136, 177
 Mean Time Between Failure · Zie MTBF
 Mean Time Between System Incidents · Zie MTBSI
 Mean Time To Repair · Zie MTTR
 meetfrequentie · 52, 110
 meetvoorschrift · 28, 52, 109, 110
 merging · 146
 meta-data · 160
 methode · 128
 methodology · 148
 metric · 146
 MFA · 177
 MI · 177

Michael Porter · 9, 10, 11
 microservice · 167
 microservice architecture · 167
 Microsoft Operations Framework · Zie MOF
 mini pipeline · 167
 Minimal Viable Product · Zie MVP
 Minimum Required Information · Zie MRI
 Misbruik · 80, 82
 mitigeren · 2, 8, 17, 28, 65, 68, 99, 140
 mock-up · 48
 Modify, Avoid, Share, Retain · Zie MASR
 Module Test · Zie MT
 MOF · 177
 monitorarchitectuur · 111, 146
 monitoring · 168
 monitorsysteem · 29
 monitorvoorziening · 18, 26, 28, 30, 34, 49, 50, 110, 111, 112, 113, 142
 monolithic · 168
 MRI · 167, 177
 MT · 177
 MTBF · 177
 MTBSI · 177
 MTTR · 168, 177
 muda · 168
 Multi Factor Authentication · Zie MFA
 MVP · 177

N

NC · 46, 50, 70, 71, 83, 87, 88, 90, 117, 118, 121, 122, 123, 124, 177
 NFR · 12, 161, 168, 177
 Non Conformity · Zie NC
 Non Functional Requirement · Zie NFR
 normenkader · 8, 22, 23, 38, 46, 62, 63, 109, 111

O

OAWOW · 177
 obeya · 168
 object code · 160
 OLA · 177
 omgeving · 133
 onderhoudbaar · 132
 One Agile Way of Working · Zie OAWOW
 one piece flow · 168
 ontwerp · 130
 ontwerper · 131
 Ontwikkel-, Test-, Acceptatie- en Productieomgeving · Zie OTAP
 ontwikkelaar · 125, 126
 ontwikkelmethode · 128
 ontwikkelproces · 24
 onzekerheidsreductie principe · 25
 Operational Level Agreement · Zie OLA
 operations · 159, 163, 168, 170, 173
 operations story · 168
 opleidingsbudget · 30
 opleidingsplan · 31

Ops engineer · 1
 Ops liaison · 168
 organisatievormgeving · 4
 organisation archetype · 168
 organisational typology model · 168
 OTAP · 177
 outcome · 7, 8, 13, 14, 17, 26, 27, 32, 33, 36, 38, 64, 69, 72, 73, 74, 75, 76, 77, 78, 79, 80, 83, 85, 86, 94, 95, 139, 140
 over-the-shoulder · 146, 169
 owner · 129, 131

P

PAAS · 177
 package · 169
 pair programming · 146, 161, 169
 password · 88, 89
 PAT · 159, 177
 patch · 3
 pattern · 160, 169
 PBI · 129, 177
 PDCA · 166, 167, 177
 peer review · 169
 peer to peer programming · 161
 People, Process & Technology · Zie PPT
 performance · 161, 167, 173, 177
 Performance StressTest · Zie PST
 performancecriterium · 48
 PESTLE · 45, 69, 75, 76, 77, 78, 177
 - categorie · 78
 - classificatie · 77
 - factor · 77, 78
 pipeline · 146, 147, 159, 165, 167, 170, 171, 173
 Plan Do Check Act · Zie PDCA
 Platform As A Service · Zie PAAS
 Political, Economic, Sociological, Technological, Legislative, Environmental · Zie PESTLE
 POP · 29
 POR · 177
 post mortem · 169
 PPT · 2, 31, 177
 PPT-aspect · 146
 principe · 22
 prioriteitscriteria · 45, 51, 83, 116
 privacy autoriteiten · 60
 probleem · 17, 18, 19
 proceseigenaar · 1
 procesmanager · 1
 Processing Time · Zie PT
 product
 - backlog · 23, 25, 26, 131, 132, 145, 149, 169
 - backlog item · 26, 165
 - functie · 131
 - owner · 1, 90, 132, 133, 169
 - portfolio · 45, 91
 Product Backlog Item · Zie PBI
 Productie AcceptatieTest · Zie PAT
 production environment · 167

programmatuur · 127, 128, 130
 programming paradigm · 169
 project · 126, 129
 Project or Organisational Risk · Zie POR
 projectteam · 132
 PSQL · 162
 PST · 177
 PT · 10, 56, 59, 61, 64, 66, 70, 75, 79, 83, 91, 94, 97, 99, 102, 104, 106, 110, 113, 116, 117, 119, 120, 122, 123, 177
 pull request process · 146, 169

Q

QA · 147, 169, 177
 QC · 177
 quality · 148
 Quality Assurance · Zie QA
 Quality Control · Zie QC

R

RACI · 25, 177
 RASCI · 25, 177
 RASCI belegging · 31
 RBAC · 177
 Real User Monitoring · Zie RUM
 recursiebeginsel · 10
 reduce batch size · 169
 reduce number of handoffs · 169
 redundantie · 23
 release · 2, 3, 138, 169
 release manager · 169
 release pattern · 169
 releaseplan · 145
 repository · 146, 147, 160, 161, 169, 170
 REpresentational State Transfer Application Programming Interface · Zie REST API
 repressie maatregel · 102
 requirement · 146, 160, 166, 168, 171, 177
 resource · 4, 22
 Responsibility, Accountable, Consulted and Informed · Zie RACI
 Responsibility, Accountable, Supporting, Consulted and Informed · Zie RASCI
 REST API · 46, 111, 113, 133, 134, 142, 150, 177
 restrisico · 46, 70, 71, 105, 106, 107
 resultaat · 128, 129
 retrospective · 163
 Return On Investment · Zie ROI
 review · 163
 richtlijn · 133
 risico · 4, 147, 160, 166
 - beheer · 71, 75
 - beoordeling · 95
 - classificatiemodel · 71, 76
 - criteria · 4
 - identificatie · 70, 71, 75, 76, 85
 - log · 18, 85

- management · 13, 40, 69
- register · 14, 46, 79, 85, 94, 96, 97, 98, 99, 101, 102, 104, 105, 107, 108, 112, 119

risk

- assessment · 4, 15, 40, 46, 69, 97, 98, 104, 135, 136
- control · 14, 46, 150
- identification · 59, 79, 84, 95, 96, 98, 135
- treatment · 4, 40, 46, 84, 94, 95, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 135, 136
- treatment option · 51, 84, 99, 100, 101, 102, 103, 104, 107, 135, 136
- treatment plan · 15, 97, 101, 105, 106, 107, 108

roadmap · 21, 25, 32, 33, 145

ROI · 177

rol · 25, 45, 46

Role-Based Access Control · Zie RBAC

rollback technique · 146

rootcause · 17

rootcause analyse · 167

RUM · 177

S

SA · 177

sad path · 169

SAFe · 177

SAFe framework · 25, 26

safety check · 169

Sarbanes Oxley · Zie SoX

SAT · 177

SBAR · 170, 177

SBB · 48, 145, 177

SBB-A · 178

SBB-I · 178

SBB-T · 48, 178

SBL · 133, 135, 136

Scaled Agile Framework · Zie SAFe

S-CI · 177

Scrum master · 1, 24, 25

Scrum ontwikkelproces · 131

Scrum team · 131, 132, 133

Secure Sockets Layer · Zie SSL

security · 161, 162, 168, 169, 170, 172

- analist · 58, 70, 71, 75, 76, 79, 83, 84, 91, 92, 94, 97, 99, 100, 102, 104, 105, 106, 107, 110, 111, 120, 121, 122, 123, 124
- best practices · 4, 25
- control · 34, 101, 111
- manager · 38, 50, 58, 90, 115, 123
- officer · 38, 50, 51, 52, 58
- policy · 40, 67

Security Acceptatie Test · Zie SAT

security officer · 162

self service capability · 170

senior management · 72

service · 177

Service Level Agreement · Zie SLA

service portfolio · 45, 91

Service Value System · Zie SVS

serviceorganisatie · 10, 70, 89, 137

shared goals · 170

shift left · 21

silo · 173

Simian army · 146, 170, 172

Simple Network Management Protocol · Zie SNMP

SIT · 178

Situation, Background, Assessment, Recommendation · Zie SBAR

skill matrix · 29

skills · 29, 31, 164

SLA · 23, 49, 71, 102, 114, 124, 147, 178, 194

SLA-normen · 23

SLA-schendingen · 71

slow feedback · 18

SM · 178

SMART · 167, 178

SME · 25, 178

smoke testing · 170

SNMP · 178

SoA · 46, 63, 101, 109, 111, 116, 178

SoE · 2, 3, 4, 24, 171, 178

software · 148, 160, 171, 173

Software Configuration Item · Zie S-CI

softwaregijzeling · 19

softwareontwikkelproces · 148

SoI · 2, 3, 4, 171, 178

SOLL · 166

SoR · 2, 3, 4, 24, 171, 178

sourcecode · 146, 160, 161, 163, 170, 171, 173

SoX · 178

Specific, Measurable, Accountable, Realistic, Timely · Zie SMART

Spotify · 25

Spotify model · 25

sprint · 163

- backlog · 132
- plan · 145
- retrospective · 129

sprint execution · 163

sprint planning · 163

SQL · 178

SRC-board · 46, 56, 57, 61, 64, 66, 114, 118, 119, 121, 123, 124

SRG · 146, 147, 178

SSL · 178

ST · 178

stakeholder · 2, 9, 14, 17, 31, 49, 50, 55, 58, 59, 85, 131, 132, 164

standard deviation · 170

standard operations · 170

Standard Rules & Guidelines · Zie SRG

stand-up · 163

Statement of Applicability · Zie SoA

statement of commitments · 57

static analysis · 171

story · 131

Strategic IS Architecture · Zie SA
 Strategic Match · Zie SM
 strategy · 148
 Structured Query Language · Zie SQL
 Subject Matter Expert · Zie SME
 super use case · 48
 SVS · 4, 7, 10, 11, 12, 33, 36, 38, 178
 systeemontwikkeling · 2
 systeemontwikkelingsproces · 125, 126
 systeemontwikkelproject · 129
 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 · 45
 System Integration Test · Zie SIT
 System of Engagement · Zie SoE
 System of Records · Zie SoR
 System Test · Zie ST
 Systems of Information · Zie SoI

T

taak · 2, 132, 159
 Taken, Verantwoordelijkheden en Bevoegdheden · Zie TVB
 target · 7, 8, 17, 18, 23, 26, 27, 33, 34, 49
 Target Operating Model · Zie TOM
 task · 165
 TCO · 178
 TCP · 178
 TDD · 146, 171, 178
 team · 127, 128, 129, 131, 132, 133
 Team Foundation Server · Zie TFS
 technical debt · 21, 148, 162, 163
 technical debt backlog · 21, 148
 Technical Information Security Officer · Zie TISO
 Technical Uncertainty · Zie TU
 technology adaption curve · 171
 technology executive · 171
 tegenmaatregel · 2, 3, 4, 8, 13, 14, 17, 40, 42, 49, 55, 64, 65, 68, 100, 101, 105, 106, 107
 telemetry · 146
 template · 47, 48, 57, 59, 77, 84, 86, 87, 88, 90, 91, 92, 94, 95, 96, 97, 98, 100, 101, 102, 105, 108, 111, 115, 116, 121, 122, 124, 135, 136, 142, 145
 test
 - case · 159, 160, 161
 - criteria · 82, 83, 84, 85, 92, 94, 96, 97, 116
 - harness · 171
 - management · 146
 Test Driven Development · Zie TDD
 testen · 25
 tester · 131, 163
 TFS · 178

The Agile Manifesto · 171
 the ideal testing automation pyramid · 172
 The Lean movement · 172
 the non-ideal testing automation inverted pyramid · 172
 The Three Ways · 168, 172
 theme · 147
 theory of constraints · 172
 Time To Market · Zie TTM
 time-box · 129, 132
 time-to-market · 126, 127
 TISO · 178
 toetsen · 25
 TOM · 7, 8, 11, 17, 28, 33, 34, 42, 50, 178
 tool-assisted code review · 146, 173
 Total Cost of Ownership · Zie TCO
 Toyota Kata · 173
 Toyota Production System · Zie TPS
 TPS · 9, 178
 traceability · 146
 traceerbaarheid · 18, 23, 143, 146, 147
 training · 132
 transactieverwerking · 24
 transformation team · 173
 Transmission Control Protocol · Zie TCP
 trigger criteria · 51, 82, 83, 84, 85, 94, 116
 trunk · 170
 T-shaped · 146, 164
 TSQL · 162
 TTM · 178
 TU · 178
 TVB · 178

U

UAT · 178
 UML · 178
 Unified Modeling Language · Zie UML
 uniformeren · 26
 Unit Test · Zie UT
 use case · 45, 46, 47, 48, 52, 57, 63, 64, 65, 67, 68, 71, 76, 79, 84, 92, 94, 95, 97, 100, 102, 105, 111, 114, 116, 118, 119, 121, 122, 124
 use case diagram · 45, 46, 47, 48
 User Acceptance Test · Zie UAT
 User eXperience design · Zie UX design
 user story · 3, 131
 UT · 131, 178
 UX design · 146, 178

V

vakmanschap · 127
 value chain · 8, 9, 10, 11, 12, 17, 26, 27, 31, 36, 48, 49, 145
 value chain model · 10
 value stream · 2, 8, 9, 10, 19, 22, 27, 32, 37, 38, 45, 46, 48, 49, 124, 145, 146, 147, 163, 167, 170, 171, 173, 178

value stream manager · 50
 Value Stream Mapping · Zie VSM
 VCR · 178
 velocity · 133, 160
 veranderparadigma · 4, 21, 22, 23, 27, 28
 verantwoordelijkheden · 4, 21
 vergadering · 132
 versie · 3
 versiebeheer · 146
 vertegenwoordiger · 128
 vertical splitting of feature · 173
 vertrouwelijkheid · 2
 Verwachte Contante ROI · Zie VCR
 virtualised environment · 173
 visibility · 146
 visie · 22
 visualisatie · 173
 Voice over Internet Protocol · Zie VOIP
 VOIP · 178
 volwassenheid · 24, 25, 151
 volwassenheidsniveau · 148
 voortbrengingsketen · 9
 voortbrengingsproces · 4
 VSM · 145, 147, 173, 178

W

waardeketen · 9, 10, 11

waardeketenactiviteiten · 9
 walking skeleton · 173
 WAN · 178
 war room · 168
 waste · 19, 26, 128, 136, 139, 140, 160, 162, 165, 166, 168, 172, 173
 waste reductie · 173
 watervalproject · 2
 Way of Working · Zie WoW
 werkvoorraad · 129
 Westrum · 168, 169
 wet- en regelgeving · 2, 8, 17, 18, 23
 Wide Area Network · Zie WAN
 wijziging · 126
 Windows Management Instrumentation · Zie WMI
 WIP · 178
 WMI · 178
 Work In Progress · Zie WIP
 workflow · 162
 WoW · 22, 178

X

XML · 178
 XML bestand · 27
 XP · 12, 128, 178

Nawoord

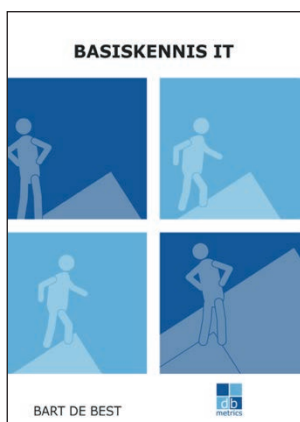
Mijn ervaring is dat de denkbeelden die ik vastleg in een artikel of een boek zich blijven evolueren. In geval u met een bepaald onderwerp uit dit boek aan de slag gaat in uw eigen DevOps organisatie, dan raad ik u aan om even met mij contact op te nemen. Wellicht zijn er aanvullende artikelen of ervaringen 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

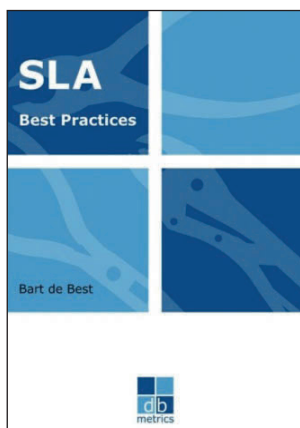
De eerste stap van een leven lang leren.

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

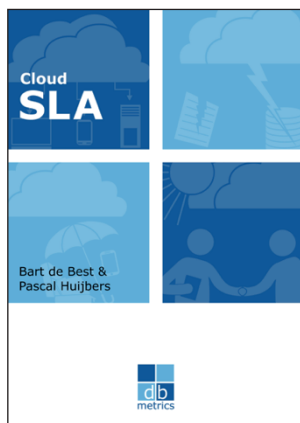
Het volledige ABC van service level agreements.

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

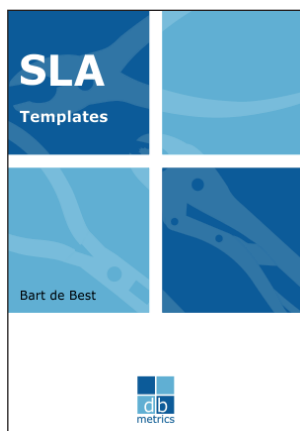
De best practices van cloud service level agreements

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-ontwikkelp proces en het beheren van de programmatuur die het Scrum-ontwikkelp proces 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



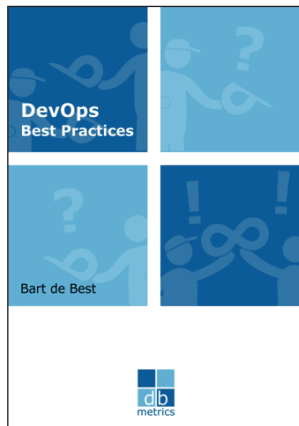
Agile Service Management met Scrum in de Praktijk

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

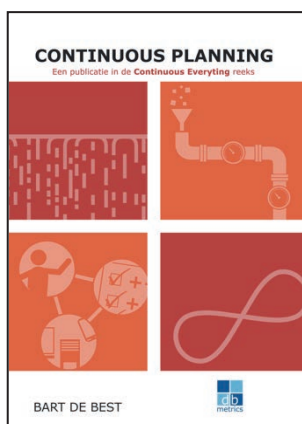
DevOps Architectuur Best Practices

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



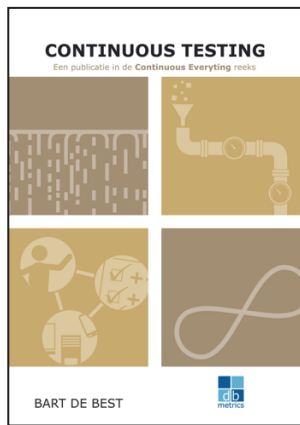
Continuous Design

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 ontwerptechnieken zoals de value stream mapping en use cases. De nadruk van het effectief toepassen van een continuous design ligt echter in de realisatie van het informatiesysteem en wel door het design te integreren in de Behaviour Driven Development en Test Driven Development alsmede in continuous documentation. Met deze Agile benadering van een design heeft u een krachtig gereedschap in handen om grip te krijgen op een Agile ontwikkelproject.

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

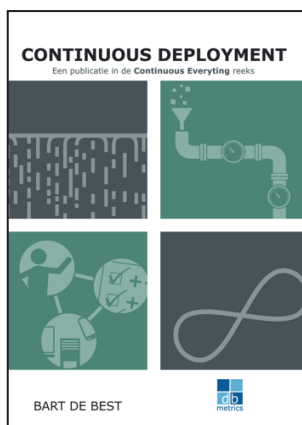
Een uitgave in de Continuous Everything reeks.

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

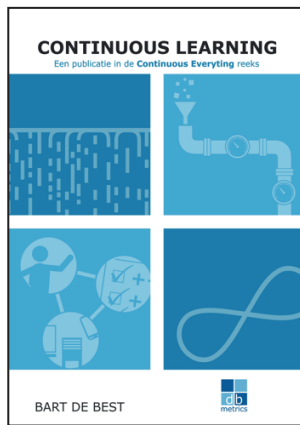
Een uitgave in de Continuous Everything reeks.

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.

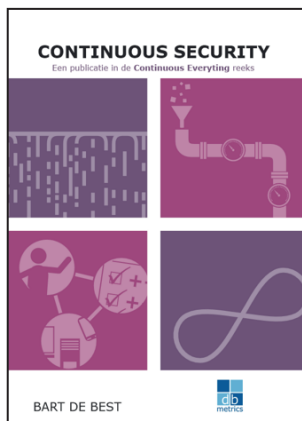
Continuous auditing is een aanpak die beoogt om DevOps teams in staat te stellen kort cyclisch aan te tonen in control te zijn bij het in hoog tempo realiseren, in productie nemen en beheren van de nieuwe of aangepaste producten en services.

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

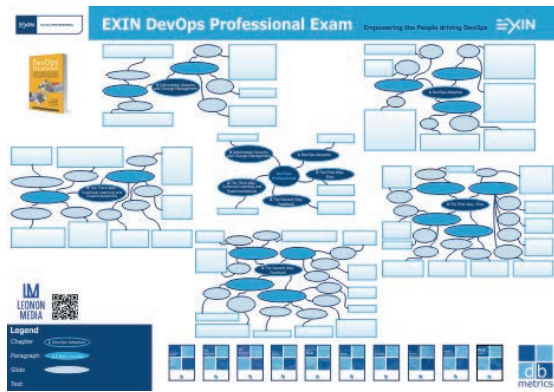
Een uitgave in de Continuous Everything reeks.

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 SECURITY

Een publicatie in de
Continuous Everything
reeks

Bart de Best



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

- 1. 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.**
- 2. Het development perspectief - Development value streams zijn in control door de non-functionele requirements voor information security integraal mee te nemen in de ontwikkeling.**
- 3. 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.

