

CONTINUOUS SERVICE INTEGRATION

Een publicatie in de **Continuous Everything** reeks



BART DE BEST



DevOps Continuous Service Integration 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

© 2024 Leonon Media

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

Titel : DevOps Continuous Service Integration
Sub titel : Een uitgave in de Continuous Everything reeks
Datum : 4 november 2024
Auteur : Bart de Best
Uitgever : Leonon Media
ISBN13 : 978 94 91480 423
Druk : Eerste druk, tweede editie, 4 november 2024

© 2024, 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..... | 3 |
| 1.4.1 | HOOFDSTUK 2: BASISCONCEPTEN EN BASISBEGRIPPEN | 3 |
| 1.4.2 | HOOFDSTUK 3: CONTINUOUS SERVICE INTEGRATION DEFINITIE..... | 3 |
| 1.4.3 | HOOFDSTUK 4: CONTINUOUS SERVICE INTEGRATION VERANKERING | 3 |
| 1.4.4 | HOOFDSTUK 5: CONTINUOUS SERVICE INTEGRATION ARCHITECTUUR | 3 |
| 1.4.5 | HOOFDSTUK 6: CONTINUOUS SERVICE INTEGRATION ONTWERP | 3 |
| 1.4.6 | HOOFDSTUK 7: PATTERN ON PREMISE SERVICE INTEGRATIE | 3 |
| 1.4.7 | HOOFDSTUK 8: PATTERN CLOUD SERVICE INTEGRATIE | 4 |
| 1.4.8 | HOOFDSTUK 9: PATTERN HYBRIDE SERVICE INTEGRATIE | 4 |
| 1.4.9 | HOOFDSTUK 10: PATTERN DEVOPS SERVICE INTEGRATIE | 4 |
| 1.4.10 | HOOFDSTUK 11: PATTERN SPECIALISATIE INTEGRATIE | 4 |
| 1.5 | BIJLAGEN | 4 |
| 1.6 | LEESWIJZER | 5 |
| 2 | BASISCONCEPTEN EN BASISBEGRIPPEN | 7 |
| 2.1 | BASISCONCEPTEN | 7 |
| 2.1.1 | SOR, SOE EN SOL..... | 7 |
| 2.1.2 | SYSTEMS OF RECORDS | 7 |
| 2.1.3 | SYSTEMS OF ENGAGEMENT..... | 7 |
| 2.1.4 | SYSTEM OF INTELLIGENCE..... | 7 |
| 2.2 | BASISBEGRIPPEN | 8 |
| 2.2.1 | ENTERPRISE ARCHITECTUUR | 8 |
| 2.2.2 | ROADMAP | 9 |
| 2.2.3 | PLANNINGSOBJECTEN | 9 |
| 2.2.4 | VALUE CHAIN..... | 10 |
| 2.2.5 | VALUE STREAM | 11 |
| 3 | CONTINUOUS SERVICE INTEGRATION DEFINITIE | 13 |
| 3.1 | ACHTERGROND | 13 |
| 3.2 | DEFINITIE..... | 13 |
| 3.3 | TOEPASSING | 13 |
| 3.3.1 | OP TE LOSSEN PROBLEMEN | 13 |
| 3.3.2 | DE ROOT CAUSE..... | 14 |
| 4 | CONTINUOUS SERVICE INTEGRATION VERANKERING..... | 17 |
| 4.1 | HET VERANDERPARADIGMA..... | 17 |
| 4.2 | BEELDVORMING | 18 |
| 4.2.1 | WAT WILLEN WE?..... | 18 |
| 4.2.2 | WAT WILLEN WE NIET? | 19 |
| 4.3 | MACHTSVERHOUDING..... | 20 |
| 4.3.1 | WAT WILLEN WE?..... | 20 |
| 4.3.2 | WAT WILLEN WE NIET? | 21 |
| 4.4 | ORGANISATIEVORMGEVING..... | 21 |
| 4.4.1 | WAT WILLEN WE?..... | 21 |
| 4.4.2 | WAT WILLEN WE NIET? | 22 |
| 4.5 | RESOURCES | 22 |
| 4.5.1 | WAT WILLEN WE?..... | 23 |

| | | |
|----------|---|-----------|
| 4.5.2 | WAT WILLEN WE NIET? | 23 |
| 5 | CONTINUOUS SERVICE INTEGRATION ARCHITECTUUR | 25 |
| 5.1 | ARCHITECTUURPRINCIPES | 25 |
| 5.1.1 | ALGEMEEN | 25 |
| 5.1.2 | PEOPLE | 25 |
| 5.1.3 | PROCESS | 26 |
| 5.1.4 | TECHNOLOGY | 28 |
| 5.2 | ARCHITECTUURMODELLEN | 29 |
| 5.2.1 | BIZDEVSECOPS.CO LEMNISCAAT | 29 |
| 5.2.2 | VIJF-PATTERN MODEL VAN CONTINUOUS SERVICE INTEGRATION | 30 |
| 6 | CONTINUOUS SERVICE INTEGRATION ONTWERP | 33 |
| 6.1 | CONTINUOUS SERVICE INTEGRATION VALUE STREAM | 33 |
| 6.2 | CONTINUOUS SERVICE INTEGRATION USE CASE DIAGRAM | 34 |
| 6.3 | CONTINUOUS SERVICE INTEGRATION USE CASE | 35 |
| 7 | PATTERN 1: ON PREMISE SI | 41 |
| 7.1 | PATTERN 1: ON PREMISE SI | 41 |
| 7.1.1 | DEFINITIE TYPE 1 | 41 |
| 7.1.2 | RISICO'S TYPE 1 | 42 |
| 7.2 | PATTERN 1A: ENKELVOUDIGE STACK SI | 43 |
| 7.2.1 | DEFINITIE SUBTYPE 1A | 43 |
| 7.2.2 | RISICO'S SUBTYPE 1A | 43 |
| 7.3 | PATTERN 1B: MEERVOUDIGE STACK SI | 43 |
| 7.3.1 | DEFINITIE SUBTYPE 1B | 43 |
| 7.3.2 | RISICO'S SUBTYPE 1B | 43 |
| 7.4 | PATTERN 1C: MEERVOUDIGE COMPONENT SI | 43 |
| 7.4.1 | DEFINITIE SUBTYPE 1C | 44 |
| 7.4.2 | RISICO'S SUBTYPE 1C | 44 |
| 7.5 | SAMENVATTING ON PREMISE SI | 44 |
| 8 | PATTERN 2: CLOUD SI | 47 |
| 8.1 | PATTERN 2: CLOUD SI | 47 |
| 8.1.1 | DEFINITIE TYPE 2 | 47 |
| 8.1.2 | RISICO'S TYPE 2 | 48 |
| 8.2 | PATTERN 2A: CLOUD LAAG SI | 49 |
| 8.2.1 | DEFINITIE SUBTYPE 2A | 49 |
| 8.2.2 | RISICO'S SUBTYPE 2A | 49 |
| 8.3 | PATTERN 2B: CLOUD KETEN SI | 50 |
| 8.3.1 | DEFINITIE SUBTYPE 2B | 50 |
| 8.3.2 | RISICO'S SUBTYPE 2B | 50 |
| 8.4 | SAMENVATTING CLOUD SI | 52 |
| 9 | PATTERN 3: HYBRIDE SI | 57 |
| 9.1 | PATTERN 3: HYBRIDE SI | 57 |
| 9.1.1 | DEFINITIE TYPE 3 | 57 |
| 9.1.2 | RISICO'S TYPE 3 | 58 |
| 9.2 | PATTERN 3A: ENKELVOUDIGE CLOUD / ON PREM SI | 58 |
| 9.2.1 | DEFINITIE SUBTYPE 3A | 58 |
| 9.2.2 | RISICO'S SUBTYPE 3A | 58 |
| 9.3 | PATTERN 3B: MEERVOUDIGE CLOUD / ON PREM SI | 58 |
| 9.3.1 | DEFINITIE SUBTYPE 3B | 58 |

| | | |
|-----------|---|------------|
| 9.3.2 | RISICO'S SUBTYPE 3B..... | 58 |
| 9.4 | SAMENVATTING HYBRIDE SI..... | 58 |
| 10 | PATTERN 4: DEVOPS SI | 59 |
| 10.1 | PATTERN 4: DEVOPS SI..... | 59 |
| 10.1.1 | DEFINITIE TYPE 4 | 59 |
| 10.1.2 | RISICO'S TYPE 4 | 59 |
| 10.2 | PATTERN 4A: GEÏSOLEERDE PLANNING SI | 61 |
| 10.2.1 | DEFINITIE SUBTYPE 4A..... | 61 |
| 10.2.2 | RISICO'S SUBTYPE 4A | 61 |
| 10.3 | PATTERN 4B: AFGESTEMDE PLANNING SI | 61 |
| 10.3.1 | DEFINITIE SUBTYPE 4B..... | 62 |
| 10.3.2 | RISICO'S SUBTYPE 4B | 62 |
| 10.4 | PATTERN 4C: GEDEELDE PLANNING SI..... | 62 |
| 10.4.1 | DEFINITIE SUBTYPE 4C..... | 62 |
| 10.4.2 | RISICO'S SUBTYPE 4C | 62 |
| 10.5 | PATTERN 4D: SELF SERVICE SI..... | 62 |
| 10.5.1 | DEFINITIE SUBTYPE 4D | 62 |
| 10.5.2 | RISICO'S SUBTYPE 4D | 63 |
| 10.6 | PATTER 4E: TEAMINTEGRATIE SI | 63 |
| 10.6.1 | DEFINITIE SUBTYPE 4E..... | 63 |
| 10.6.2 | RISICO'S SUBTYPE 4E..... | 63 |
| 10.7 | SAMENVATTING DEVOPS SI..... | 63 |
| 11 | PATTERN 5: SPECIALISATIE SI | 67 |
| 11.1 | PATTERN 5: SPECIALISATIE SI | 67 |
| 11.1.1 | DEFINITIE TYPE 5 | 67 |
| 11.1.2 | RISICO'S TYPE 5 | 68 |
| 11.2 | PATTERN 5A: BUSINESS SPECIALISATIE SI | 69 |
| 11.2.1 | DEFINITIE SUBTYPE 5A..... | 69 |
| 11.2.2 | RISICO'S SUBTYPE 5A | 69 |
| 11.3 | PATTERN 5B: APPLICATIE SPECIALISATIE SI | 69 |
| 11.3.1 | DEFINITIE SUBTYPE 5B..... | 69 |
| 11.3.2 | RISICO'S SUBTYPE 5B | 69 |
| 11.4 | PATTERN 5C: MICROSERVICE SPECIALISATIE SI..... | 70 |
| 11.4.1 | DEFINITIE SUBTYPE 5C..... | 70 |
| 11.4.2 | RISICO'S SUBTYPE 5C | 70 |
| 11.5 | PATTERN 5D: HYBRIDE SI..... | 70 |
| 11.5.1 | DEFINITIE SUBTYPE 5D | 70 |
| 11.5.2 | RISICO'S SUBTYPE 5D | 70 |
| 11.6 | SAMENVATTING DEVOPS SPECIALISATIE SI | 71 |
| 12 | CONTINUOUS SERVICE INTEGRATION ASSESSMENT..... | 75 |
| 12.1 | WAT IS HET CE-MODEL | 75 |
| 12.2 | VOLWASSENHEIDSDIMENSIES | 78 |
| 12.3 | DEVOPS CE MODEL, CR..... | 79 |
| | BIJLAGE A, LITERATUURLIJST | 85 |
| | BIJLAGE B, BEGRIPPENLIJST | 89 |
| | BIJLAGE C, AFKORTINGEN..... | 105 |
| | BIJLAGE D, WEBSITES..... | 111 |

BIJLAGE E, INDEX 113

Figuren

| | |
|--|----|
| FIGUUR 1-1, BizDevSecOps.CO LEMNISCAAT. | 1 |
| FIGUUR 1-2, CONTINUOUS SERVICE INTEGRATION PATTERNS..... | 2 |
| FIGUUR 2-1, SOR, SOE EN SOI (BRON: HSO THE RESULT COMPANY). | 8 |
| FIGUUR 2-2, ENTERPRISE ARCHITECTURE. | 9 |
| FIGUUR 2-3, ROADMAP. | 9 |
| FIGUUR 2-4, VALUE CHAIN OF PORTER, BRON: [BOEK MICHAEL PORTER]. | 10 |
| FIGUUR 4-1, VERANDERPARADIGMA. | 17 |
| FIGUUR 4-2, VERANDERPARADIGMA - BEELDVORMING..... | 18 |
| FIGUUR 4-3, VERANDERPARADIGMA - MACHTSVERHOUDING..... | 20 |
| FIGUUR 4-4, VERANDERPARADIGMA - ORGANISATIE. | 21 |
| FIGUUR 4-5, VERANDERPARADIGMA - RESOURCES. | 23 |
| FIGUUR 5-1, BizDevSecOps.CO LEMNISCAAT. | 29 |
| FIGUUR 5-2, CONTINUOUS SERVICE INTEGRATION MODEL. | 30 |
| FIGUUR 6-1, BizDevSecOps.CO LOGO. | 33 |
| FIGUUR 6-2, CONTINUOUS SERVICE INTEGRATION VALUE STREAM..... | 34 |
| FIGUUR 6-3, USE CASE DIAGRAM VOOR CONTINUOUS SERVICE INTEGRATION. | 34 |
| FIGUUR 7-1, PATTERN TYPE 1: ON PREMISE SERVICE INTEGRATIE. | 41 |
| FIGUUR 8-1, TYPE 2: CLOUD SERVICE INTEGRATION. | 47 |
| FIGUUR 9-1, TYPE 3: PATTERN CLOUD / ON PREM SERVICE. | 57 |
| FIGUUR 10-1, TYPE 4: DEVOPS SERVICE INTEGRATIE..... | 59 |
| FIGUUR 11-1, TYPE 5: DEVOPS SPECIALISATIE SERVICE INTEGRATIE. | 67 |
| FIGUUR 12-1, DEVOPS CE-SPIDER MODEL..... | 78 |
| FIGUUR 12-2, DEVOPS CR-SPIDER MODEL..... | 81 |

Tabellen

| | |
|---|----|
| TABEL 1-1, CONTINUOUS EVERYTHING ASPECTEN. | 2 |
| TABEL 1-2, BIJLAGEN. | 5 |
| TABEL 2-1, PLANNINGSOBJECTEN. | 10 |
| TABEL 3-1, VEEL VOORKOMENDE PROBLEMEN BIJ HET HANTEREN VAN EEN DESIGN. | 14 |
| TABEL 5-1, BizDevSecOps.CO EN DE GERELATEERDE CE VALUE STREAMS..... | 30 |
| TABEL 6-1, USE CASE TEMPLATE. | 35 |
| TABEL 6-2, USE CASE VOOR CONTINUOUS SERVICE INTEGRATION. | 39 |
| TABEL 7-1, RELATIE CONTINUOUS EVERYTHING EN DE RISICO'S VAN DE ON PREMISE SERVICE INTEGRATIE..... | 46 |
| TABEL 8-1, RELATIE CONTINUOUS EVERYTHING EN DE RISICO'S VAN DE CLOUD SERVICE INTEGRATIE. | 55 |
| TABEL 10-1, RELATIE CONTINUOUS EVERYTHING EN DE RISICO'S VAN DE DEVOPS SERVICE INTEGRATIE..... | 65 |
| TABEL 11-1, RELATIE CONTINUOUS EVERYTHING EN DE RISICO'S VAN DE DEVOPS SPECIALISATIE SI. | 73 |
| TABEL 12-1, DEVOPS CE-MODEL. | 75 |
| TABEL 12-2, CONTINUOUS EVERYTHING. | 76 |
| TABEL 12-3, CMMI LEVELS VOOR CONTINUOUS EVERYTHING. | 77 |
| TABEL 12-4, PR-ORG-009. VOLWASSENHEIDSNIVEAUS. | 78 |
| TABEL 12-5, CR MATURITY CHARACTERISTICS. | 80 |

Bijlagen

| | |
|----------------------------------|-----|
| BIJLAGE A, LITERATUURLIJST | 85 |
| BIJLAGE B, BEGRIPPENLIJST | 89 |
| BIJLAGE C, AFKORTINGEN | 105 |
| BIJLAGE D, WEBSITES | 111 |
| BIJLAGE E, INDEX | 113 |

Voorwoord

Dit boek is samengesteld op basis van mijn ervaringen bij het ontwerpen van geïntegreerde services in zowel DevOps als cloud omgevingen. 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 Service Integration, 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 Service Integration 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 geïntegreerde services van toepassing zullen zijn en dat de aanpak een momentopname is die wellicht door de toenemende snelheid van innovatie weer achterhaald kan zijn.

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

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

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

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

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

1 Introductie

Leeswijzer:

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

1.1 Doel

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

1.2 Doelgroep

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

1.3 Achtergrond

Dit boek bevat verschillende technieken om op een continue wijze invulling te geven aan een geïntegreerde service die meegroeit met de business door deze samen met verschillende service providers te ontwikkelen en te leveren. Continuous Service Integration is een onderdeel van Continuous Everything. In [Figuur 1-1](#) is dit aspectgebied van Continuous Everything afgebeeld als een ring om alle andere Continuous Everything aspectgebieden omdat de integratie van services niet zozeer een stap is in de BizDevSecOps Lemniscaat maar een integrale besturing.



Figuur 1-1, BizDevSecOps.Co Lemniscaat.

Continuous Service Integration is één van de vele aspectgebieden van het Continuous Everything concept zoals gedefinieerd binnen het BizDevSecOps.Co Lemniscaat. Het CE-concept beschrijft alle fasen van [Figuur 1-1](#) in de vorm van continue uit te voeren activiteiten. In [Tabel 1-1](#) is de relatie weergegeven tussen de stappen van de DevOps lemniscaat en de Continuous Everything aspectgebieden.

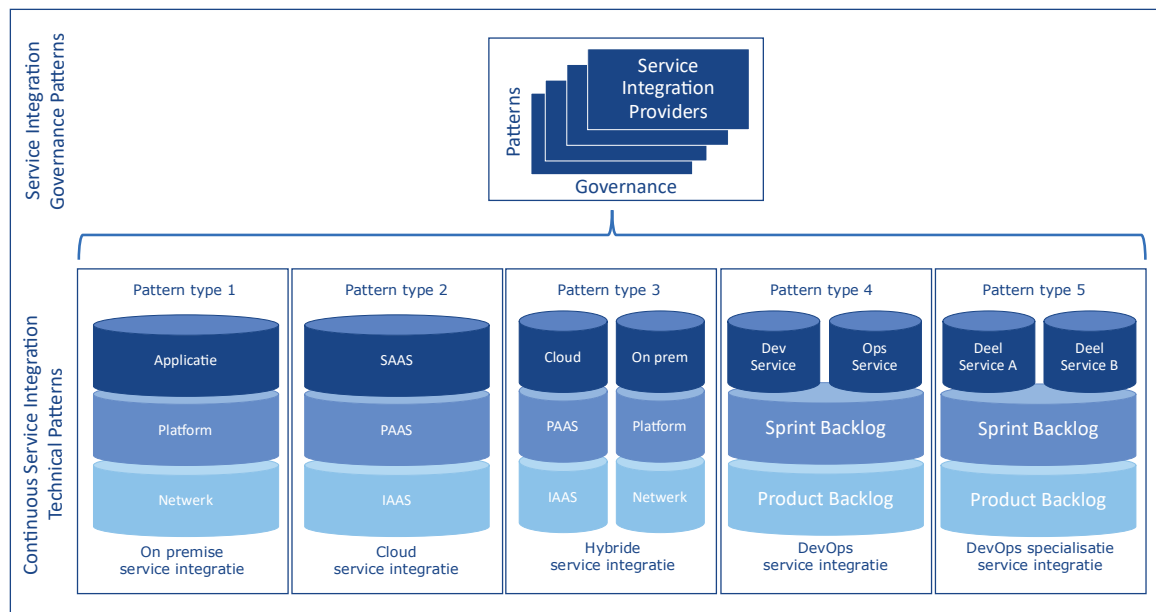
| Biz - Business | | Dev - Development | |
|----------------|--------------------------------|-------------------|-------------------------------------|
| CQ | Continuous SLA (Agree) | CH | Continuous Architecture (Architect) |
| CC | Continuous Acceptance (Accept) | CP | Continuous Planning (Plan) |
| CO | Continuous Outcome (Outcome) | CN | Continuous Design (Design) |
| | | CT | Continuous Testing (Test) |
| | | CI | Continuous Integration (Code) |
| | | CZ | Continuous AI (AI) |
| Sec - Security | | Ops - Operations | |
| CY | Continuous Security (Secure) | CD | Continuous Deployment (Deploy) |
| CS | Continuous Assessment (Assess) | CD | Continuous Deployment (Release) |
| CA | Continuous Auditing (Audit) | CM | Continuous Monitoring (Monitor) |
| | | CL | Continuous Learning (Learn) |

Tabel 1-1, Continuous Everything aspecten.

Continuous Service Integration beschrijft op welke wijze een service continue kan worden vormgegeven, aangepast en bestuurd waarbij meer service providers hecht samenwerken. Hierbij worden in dit boek verschillende patronen (patterns) van service integratie onderkend in dit boek te weten:

1. Pattern Type 1: On premise service integratie
2. Pattern Type 2: Cloud service integratie
3. Pattern Type 3: Hybride service integratie
4. Pattern Type 4: DevOps service integratie
5. Pattern Type 5: Specialisatie service integratie

In [Figuur 1-2](#) is een overzicht gegeven van de samenhang van deze service integration patterns.



Figuur 1-2, Continuous Service Integration patterns.

De bovenkant van [Figuur 1-2](#) toont de besturing patterns (governance) van de Continuous Service Integration pattern.

De onderkant geeft de vijf service integration patterns weer die in dit boek worden besproken. De governance patterns geven weer op welke wijze de functie van service integrator kunnen worden toegepast in een organisatie. Het uitgangspunt is om de business de ervaring te geven dat de service slechts door één service provider, de service integrator wordt geleverd. Hierdoor wordt de business ontlast van de governance van de onderliggende service providers. Een veel gebruikt governance model hiertoe is Service Integration And Management (SIAM). Hierbij worden vier governance patterns gehanteerd te weten:

- een interne service integrator die de klant zelf invult;
- een externe service integrator waarbij de integrator rol wordt uitbesteed;
- een lead supplier aansturing waarbij één van de service providers de integrator is;
- een hybride model waarbij de klant en externe service integrator samenwerken.

Dit boek bespreekt de belegging van de service integrator rol niet. Wel worden de patterns (typen service integraties) besproken die gebruikt kunnen worden om te komen tot een service integratie. De service providers kunnen zowel intern als extern zijn. En alle vier de governance patterns kunnen hierbij worden gebruikt. Voor elke service integratie patterns worden de volgende onderwerpen besproken:

- de definitie
- de risico's
- de subtypen
- de definitie van de subtypen
- de risico's van de subtypen
- de toepassing van BizDecSecOps.Co op de risicobeheersing

In hoofdstuk 5 zijn de patterns beschreven.

1.4 Structuur

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

1.4.1 Hoofdstuk 2: Basisconcepten en basisbegrippen

Dit hoofdstuk bespreekt de basisconcepten en de basisbegrippen.

1.4.2 Hoofdstuk 3: Continuous Service Integration definitie

Het is belangrijk om een gemeenschappelijke definitie te hebben van Continuous Service Integration. Daarom wordt er in dit hoofdstuk een definitie gegeven van dit concept en worden de problemen en mogelijke oorzaken van het niet goed samenstellen van een geïntegreerde service besproken.

1.4.3 Hoofdstuk 4: Continuous Service Integration verankering

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

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

1.4.4 Hoofdstuk 5: Continuous Service Integration Architectuur

Dit hoofdstuk beschrijft de architectuur voor de toepassing van Continuous Service Integration op basis van het BizDevSecOps architectuurmodel.

1.4.5 Hoofdstuk 6: Continuous Service Integration Ontwerp

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

1.4.6 Hoofdstuk 7: Pattern on premise service integratie

Deze pattern omvat de geïntegreerde serviceverlening aan een klant die door een samenwerking van meer service providers wordt geleverd op basis van een lokale installatie en exploitatie van een applicatie en de vereiste stack bij de klant (on premise).

De volgende subtypen van deze pattern worden onderkend:

- Meervoudige stack service integratie
- Multi component service integratie

1.4.7 Hoofdstuk 8: Pattern cloud service integratie

Deze pattern omvat de geïntegreerde serviceverlening aan een klant die door een samenwerking van cloud service providers wordt geleverd op basis van een installatie en exploitatie van een applicatie en de vereiste stack in de cloud.

Hierbij worden de volgende subtypen onderkend:

- Pattern cloud service integratie
- Cloud keten service integratie

1.4.8 Hoofdstuk 9: Pattern hybride service integratie

Deze pattern omvat de geïntegreerde serviceverlening aan een klant die door een samenwerking van meer service providers wordt geleverd op basis van zowel een cloud service als een on premise service.

De volgende subtypen van deze pattern worden onderkend:

- Hybride service integratie
- Enkelvoudige cloud / on premise service
- Meervoudige cloud / on premise service

1.4.9 Hoofdstuk 10: Pattern DevOps service integratie

Deze pattern omvat de geïntegreerde serviceverlening aan een klant die door een samenwerking van DevOps engineers wordt geleverd op basis van een cross functioneel team.

Hierbij worden de volgende subtypen onderkend:

- Geïsoleerde planning service integratie
- Afgestemde planning service integratie
- Gedeelde planning service integratie
- Self service service integratie
- Team integratie service integratie

1.4.10 Hoofdstuk 11: Pattern specialisatie integratie

Deze pattern omvat de geïntegreerde serviceverlening aan een klant die door een samenwerking van gespecialiseerde DevOps teams wordt geleverd. Hierbij worden de volgende subtypen onderkend:

- Business specialisatie integratie
- Applicatie specialisatie integratie
- Microservice specialisatie integratie
- Hybride specialisatie integratie

1.5 Bijlagen

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

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

| Bijlagen | Onderwerp | Toelichting |
|----------|-----------|--|
| E | Websites | Een aantal relevante websites zijn in deze bijlage opgenomen. In dit boek wordt verwezen naar deze websites door de referentie: [http Name]. |
| F | Index | De index omvat de verwijzing van termen die in dit boek zijn gehanteerd. |

Tabel 1-2, Bijlagen.

1.6 Leeswijzer

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

Bijlagen

Bijlage A, Literatuurlijst

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

| Referenties | Publicaties |
|--------------|--|
| Best 2011a | B. de Best, "SLA best practices", Leonon Media 2011, ISBN13: 978 90 71501 456. |
| Best 2011b | B. de Best, "ICT Performance-Indicatoren", Leonon Media 2011, ISBN13: 978 90 71501 470. |
| Best 2012 | B. de Best, "Quality Control & Assurance", Leonon Media 2012, ISBN13: 978 90 71501 531. |
| Best 2014a | B. de Best, "Acceptatiecriteria", Leonon Media, 2014, ISBN 13: 978 90 71501 784. |
| Best 2014b | B. de Best, "Agile Service Management met Scrum", Leonon Media, 2014, ISBN13: 978 90 71501 807. |
| Best 2014c | B. de Best, "Cloud SLA, Leonon Media, 2014 ISBN13: 978 90 71501 739. |
| Best 2015a | B. de Best, "Agile Service Management met Scrum in de Praktijk", Leonon Media, 2015, ISBN13: 978 90 71501 845. |
| Best 2017a | B. de Best, "Beheren onder Architectuur", Dutch language, Leonon Media, 2017, ISBN13: 978 90 71501 913. |
| Best 2017b | B. de Best, "DevOps best practices", English language, Leonon Media, 2017, ISBN13: 978 94 92618 078. |
| Best 2017c | B. de Best, "SLA Templates", English language, Leonon Media, 2017, ISBN13: 978 94 92618 320. |
| Best 2019b | B. de Best, "DevOps Architectuur", Dutch language, Leonon Media, 2019, ISBN13: 978 94 92618 061. |
| Best 2021a | 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 2023 CC | B. de Best, "Continuous Acceptance", Dutch language, Leonon Media, 2023, ISBN13: 978 94 91480 317. |
| Best 2022 CD | B. de Best, "Continuous Deployment", Dutch language, Leonon Media, 2022, ISBN13: 978 94 92618 511. |
| Best 2022 CH | B. de Best, "Continuous Architecture", Dutch language, Leonon Media, 2024, ISBN13: 978 94 91480 348. |
| 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 2024 CO | B. de Best, "Continuous Outcome", Dutch language, Leonon Media, 2024, ISBN13: 978 94 91480 386. |
| Best 2022 CP | B. de Best, "Continuous Planning", Dutch language, Leonon Media, 2022, ISBN13: 978 94 92618 504. |

| Referenties | Publicaties |
|----------------|--|
| Best 2023 CQ | B. de Best, "Continuous SLA", Dutch language, Leonon Media, 2023, ISBN13: 978 94 91480 263. |
| Best 2024 CR | B. de Best, "Continuous Service Integration", Dutch language, Leonon Media, 2024, ISBN13: 978 94 91480 423. |
| 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. |
| Best 2022 CY | B. de Best, "Continuous Security", Dutch language, Leonon Media, 2022, ISBN13: 978 94 91480 171. |
| Best 2023 CZ | B. de Best, "Continuous AI", Dutch language, Leonon Media, 2023, ISBN13: 978 94 91480 294. |
| 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. |
| Best 2023 | B. de Best, "Continuous Everything een Introductie", Dutch language, Leonon Media, 2023, ISBN13: 978 94 91480 270. |
| Best 2024 | B. de Best, "Continuous Business", Dutch language, Leonon Media, 2024, ISBN13: 978 94 91480 362 |
| 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, |

| Referenties | Publicaties |
|---------------------|--|
| | 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. |
| 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'. |
| 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. The colour green refers to the situation that a factory is built on a clean grass field. |

| Begrip | Betekenis |
|---------------------------------------|--|
| | 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) proofs 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. |
| Compliance checking | The manual action of a security officer to make sure that the system is built in accordance with the agreed standards. |

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

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

| Begrip | Betekenis |
|--|---|
| | 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. • Estimable: The size of a product backlog item must always estimable. |

| Begrip | Betekenis |
|------------------------------|---|
| | <ul style="list-style-type: none"> • 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) | <p>JKK which means 100% completion of an item. This quality way of working means:</p> <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; <p>and then maintaining the required quality without inspections.</p> |
| 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. This can be accomplished by finding the root cause of a failure by asking "Why" 5 times.</p> |
| | <p>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) | <p>A Kaizen Blitz is a rapid improvement workshop designed to produce results / approaches to discrete process issues within a few days.</p> <p>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.</p> |
| Kaizen in advance | <p>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.</p> |
| 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 | <p>A Kibana dashboard displays a collection of saved visualisations.</p> |
| Latent defects | <p>Problems that are not visible yet. Latent defects can be made visible by injecting faults into the system.</p> |
| Launch Readiness Review (LRR) | <p>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.</p> |
| Launching guidance | <p>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].</p> |
| Lead Time (LT) | <p>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.</p> |

| Begrip | Betekenis |
|------------------------------|--|
| Lean tools | <ul style="list-style-type: none"> • A3 thinking (problem solving) • Continuous flow (eliminates waste) • <u>Kaizen</u> • <u>Kanban</u> • KPI (Key Performance Indicator) • Plan Do Check Act (PDCA) • Root cause analysis • Specific, Measurable, Accountable, Realistic, Timely (SMART) • <u>Value stream mapping</u> (depict the flow) • <u>JKK</u> (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 (<u>ITSM</u>) 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. |
| 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] . |

| Begrip | Betekenis |
|-------------------------------|--|
| 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. |
| 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 runtime 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.</p> <p>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. |

| Begrip | Betekenis |
|---|--|
| | 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 | <p>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.</p> <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 | 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. |
| 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.</p> <p>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. 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); |

| Begrip | Betekenis |
|--------------------------------|--|
| | <ul style="list-style-type: none"> • 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 | 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. |
| 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. |

| Begrip | Betekenis |
|------------------------|--|
| 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. |
| 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 |
| ASL | Application Services Library |
| AVG | Algemene Verordening Gegevensbescherming |
| AWS | Amazon Web Services |
| BDD | Behavior Driven Development |
| BI | Business Intelligence |
| BiSL | Business Information Services Library |
| BOK | Body of Knowledge |
| BSC | Balanced ScoreCard |
| BVS | Business Value System |
| CA | Competitive Advantage |
| CAB | Change Advisory Board |
| CAMS | Culture, Automation, Measurement and Sharing |
| CE | Continuous Everything |
| CE CA | Continuous Auditing |
| CE CC | Continuous Acceptance |
| CE CD | Continuous Deployment |
| CE CE | Continuous Everything |
| CE CH | Continuous Architecture |
| CE CI | Continuous Integration |
| CE CL | Continuous Learning |
| CE CM | Continuous Monitoring |
| CE CN | Continuous design |
| CE CO | Continuous Outcome |
| CE CP | Continuous Planning |
| CE CR | Continuous Service Integration |
| CE CS | Continuous aSessment |
| CE CT | Continuous Testing |
| CE CQ | Continuous SLA |
| CE CY | Continuous Security |
| CE CZ | Continuous AI |
| CEM | Central Event Monitor |
| CEMLI | Configuration, Extension, Modification, Localisation, Integration |
| CEO | Chief Executive Officer |
| CFO | Chief Finance Officer |
| CI | Configuration Item |
| CIA | Confidentiality, Integrity & Availability |
| CIO | Chief Information Officer |
| CMDB | Configuration Management DataBase |
| CMMI | Capability Maturity Model Integration |
| CMS | Configuration Management System |

| Afkorting | Betekenis |
|-----------|--|
| CO | Continuous dOcumentation |
| CoC | Code of Conduct |
| CoP | Communities of Practice (CoP) |
| CPU | Central Processing Unit |
| CR | Competitive Response |
| CRAMM | CCTA Risk Assessment Method Methodology |
| CRC | Cyclic Redundancy Check |
| CTO | Chief Technical Officer |
| DevOps | Development & Operations |
| DL | Deep Learning |
| 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 |

| Afkorting | Betekenis |
|-----------|--|
| ISO | Information Standardisation Organisation |
| ISVS | Information Security Value System |
| IT | Information Technology |
| ITIL 4 | Information Technology Infrastructure Library 4 |
| ITSM | Information Technology Service Management |
| JIC | Just In Case |
| JIT | Just In Time |
| JKK | Ji-Kotei-Kanketsu |
| JVM | Java Virtual Machine |
| KPI | Key Performance Indicator |
| KSF | Kritieke Succes Factor |
| 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 |
| ML | Machine Learning |
| 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 |

| Afkorting | Betekenis |
|-----------|---|
| 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 |
| RL | Reinforcement Learning |
| RPA | Robotic Process Automation |
| 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 |
| 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 |
| SRC | Security, Risk and Compliancy |
| SRG | Standards Rules & Guidelines |
| SSL | Secure Sockets Layer |
| ST | Systeemtest |
| SVS | Service Value System |
| SWOT | Strength, Weakness, Opportunities, Threats |
| 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 |

| Afkorting | Betekenis |
|-----------|--|
| 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/ |
| BizDevSecOps | [BizDevSecOps] | http://bizdevsecops.co |
| Continuous Everything NL | [CE NL] | http://www.continuouseverything.nl |
| Continuous Everything UK | [CE UK] | http://www.continuouseverything.org |
| dbmetrics | [dbmetrics] | http://www.dbmetrics.nl |
| dbmetrics | [dbmetrics publicaties] | https://www.leonon.nl/book/CUSTBestPracticePublications-best-practice-publications.html |
| 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#a0525f798cda415a94fedecb806d2c49 |
| DutchNordicGroup | [DNG] | https://www.dutchnordic.group |
| EXIN | [Exin] | http://www.exin.nl |
| ITMG | [ITMG] | http://www.ITMG.nl |
| ITPedia | [ITPEDIA] | http://www.itpedia.nl |
| Tiobe | [Tiobe] | www.tiobe.com/content/paperinfo/DefinitionOfConfidenceFactor.html |
| 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 · 105

A

A/B testing · 76, 89
 acceptatiecriterium · 90, 93
 acceptatietest · 89
 actor · 35, 37
 affinity · 89
 AFM · 105
 Agile · 4, 77, 89, 102
 - coach · 1
 - design · 7
 - infrastructure · 89
 - proces · 77
 - project · 10
 - Scrum · 93
 Algemene Verordening
 Gegevensbescherming · Zie AVG
 alternate path · 89
 Amazon Web Services · Zie AWS
 Andon cord · 89
 anker · 17
 anomaly detection technique · 89
 antifragility · 90
 anti-pattern · 18, 19, 21, 22, 23, 90
 apenrots · 17
 applicatie · 76
 - architectuur · 8
 - beheer · 96
 - component · 97
 Application Services Library · Zie ASL
 architectuur · 8
 - framework · 8
 - model · 8
 - principe · 8
 - view · 8
 artefact · 34, 90, 92
 artefact repository · 90
 ASL · 105
 assessment · 15, 36, 100
 auditing · 76
 auditor · 1
 automated test · 90
 Autoriteit Financiële Markten · Zie AFM
 availability · 91
 AVG · 7, 105
 AWS · 105

B

backlog item · 99
 bad apple theory · 90

Balanced ScoreCard · Zie BSC
 base use case · 35
 baseline · 76
 BDD · 76, 90, 105
 bedrijfsproces · 76
 beeldvorming · 3, 17, 18, 19, 20, 21, 22
 Behavior Driven Development · Zie BDD
 best practice · 17, 78, 91
 beveiliging · 77
 bevoegdheden · 3, 17
 BI · 105
 BI service · 14
 BI-model · 7
 binary · 90
 BI-oplossing · 7
 BiSL · 105
 BizDevSecOps architectuurmodel · 3
 BizDevSecOps assessment · 23
 BizDevSecOps.Co Lemniscaat · 29
 blameless post mortem · 76, 90
 blamelessness · 90
 blue/green deployment · 90
 blue/green environment · 76
 Body of Knowledge · Zie BOK
 BOK · 105
 bottom-up · 17
 branching · 76, 91
 broken build · 76, 90
 brown field · 91
 BSC · 105
 build · 76, 90, 91, 92, 101
 build-in failure mode · 76
 business
 - architectuur · 8
 - DevOps · 76
 - value · 91, 93
 Business Information Services Library · Zie
 BiSL
 Business Intelligence · Zie BI
 Business Value System · Zie BVS
 BVS · 105

C

CA · 105
 CAB · 105
 CAMS · 92, 105
 canary releasing · 76, 91
 capability · 92
 Capability Maturity Model Integration · Zie
 CMMI
 capaciteit · 91
 CCTA Risk Assessment Method
 Methodology · Zie CRAMM
 CD · 76, 78
 CE · 78, 105
 CE assessment · 39
 CE audit · 39
 CE CA · 105

- CE CC · 105
- CE CD · 91, 95, 105
- CE CH · 105
- CE CI · 91, 95, 105
- CE CL · 105
- CE CM · 105
- CE CN · 105
- CE CO · 105, 106
- CE CP · 105
- CE CQ · 105
- CE CS · 105
- CE CT · 105
- CE CY · 105
- CE CZ · 105
- CE value stream · 29
- CEM · 105
- CEMLI · 105
- CE-model · 75
- Central Event Monitor · Zie CEM
- Central Processing Unit · Zie CPU
- CEO · 105
- CFO · 105
- Change Advisory Board · Zie CAB
- change category · 91
- change schedule · 91
- check-in · 76
- checkpunt · 80
- Chief Executive Officer · Zie CEO
- Chief Finance Officer · Zie CFO
- Chief Information Officer · Zie CIO
- Chief Technology Officer · Zie CTO
- CI · 75, 76, 78, 105
- CI/CD secure pipeline · 42, 65
- CI/CD secure pipeline service · 31
- CIA · 105
- CIA matrix · 60
- CIO · 105
- CL · 76
- cloud · 91
- cloud configuration file · 91
- cloud service · 91
- cluster immune system release pattern · 91
- CM · 76, 79, 80
- CMDB · 105
- CMMI · 75, 77, 78, 79, 105
- CMS · 105
- CoC · 106
- code branch · 91
- Code of Conduct · Zie CoC
- code review form · 91
- codified NFR · 91
- collaboration · 91
- commit code · 91
- commit stage · 91
- Communities of Practice · Zie CoP
- competence · 89, 94
- Competitive Advantage · Zie CA
- Competitive Response · Zie CR
- Completeness / Accurateness · Zie %C/A
- compliance · 76, 92
- compliance checking · 92
- compliance officer · 77, 92
- compliance officer · 92
- component · 95, 98, 101
- concurrentievoordeel · 10
- confidentiality · 105
- Confidentiality, Integrity & Availability · Zie CIA
- configuratiebeheer · 76
- Configuration Item · Zie CI
- configuration management · 92
- Configuration Management DataBase · Zie CMDB
- Configuration Management System · Zie CMS
- Configuration, Extention, Modification, Localisation, Integration · Zie CEMLI
- constructiefout · 8
- container · 92
- continuity · 91, 97
- Continuous
 - Assessment · 4, 13, 18, 19, 25, 26, 28, 34, 36, 41, 57, 59
 - Delivery · 78
 - Deployment · 2
 - Design · 2
 - Everything · 1, 2, 13, 75, 76, 77, 78
 - Integration · 2
 - Learning · 2, 103
 - Monitoring · 2
 - Planning · 2
 - Testing · 2
- Continuous Acceptance · Zie CE CA
- Continuous AI · Zie CE CZ
- Continuous Architecture · Zie CE CH
- Continuous aSessment · Zie CE CS
- Continuous Auditing · Zie CE CA
- Continuous Deployment · Zie CE CD
- Continuous design · Zie CE CN
- Continuous dOcumentation · Zie CO
- Continuous Everything · See CE, Zie CE CE
- Continuous Everything Academy · 61
- continuous improvement · 103
- Continuous Integration · Zie CE CI
- Continuous Learning · Zie CL
- Continuous Monitoring · Zie CE CM
- Continuous Outcome · Zie CE CO
- Continuous Planning · Zie CE CP
- Continuous Security · Zie CE CY
- Continuous Service Integration · 13, 14
- Continuous SLA · 23, Zie CECQ
- Continuous Testing · Zie CE CT
- Continuous Testing roadmap · 14
- control · 78, 92, 100
- Conway's law · 92
- CoP · 106
- counter measure · 96
- CPU · 106
- CR · 106
- CRAMM · 106
- CRC · 106
- CT · 76
- CTO · 106
- cultural debt · 92
- Culture, Automation, Measurement and Sharing · Zie CAMS

cultuur · 14
 cycle time · 76, 92
 Cyclic Redundancy Check · Zie CRC

D

data · 78
 datakwaliteit · 19
 data-reconciliatie · 45
 data-stewardship · 19
 data-vierkantscontrole · 45
 debt · 92
 declarative programming · 92
 Deep Learning · See DL
 defect · 77, 97
 defect tracking · 93
 Definition of Done · Zie DoD
 Definition of Ready · Zie DoR
 Definitional Uncertainty · Zie DU
 Definitive Media Library · Zie DML
 Demming wheel · 96
 deployment · 89
 deployment pipeline · 91
 design · 92, 104
 Dev engineer · 1
 development · 89, 90, 93, 95, 96, 98, 100, 101, 102, 104
 Development & Operations · Zie DevOps
 development model · 102
 development ritual · 93
 Development Value System · Zie DVS
 Development, Test, Acceptance and Production · Zie DTAP
 DevOps · 7, 17, 75, 77, 78, 89, 91, 93, 99, 103, 106
 - capabilities · 14
 - engineer · 89, 91, 92, 93, 100, 102
 - team · 1, 89, 90, 91, 92, 94, 98, 103
 - werkwijze · 1
 DL · 106
 DML · 106
 DNS · 106
 DoD · 37, 77, 95, 106
 Domain Name System · Zie DNS
 DoR · 77, 106
 downward spiral · 93
 DTAP · 100, 106
 DTAP environments · 100
 DU · 106
 DVS · 106

E

E2E · 77, 106
 economics of scales · 13
 eigenaarschap · 14, 17
 e-mail pass around · 76, 93
 empirisch · 25
 End User eXperience Monitoring · Zie EUX
 End-to-End · Zie E2E
 Enterprise Architectuur model · 8

enterprise architectuurmodel · 8
 Enterprise Resource Planning · Zie ERP
 Enterprise Service Buss · Zie ESB
 Entity Relation Diagram · Zie ERD
 epic · 9, 10, 77, 106
 Epic Solution Approach · Zie ESA
 ERD · 106
 ERP · 106
 error path · 93
 ESA · 106
 ESB · 106
 E-shaped · 76, 94
 ETL · 106
 EUX · 106
 event · 76, 98
 exit-strategie · 19
 eXtensible Markup Language · Zie XML
 Extract Transform & Load · Zie ETL
 eXtreme Programming · Zie XP

F

failure · 90
 FAT · 89, 106
 feature · 9, 10, 77, 93, 94, 103
 Feature Solution Approach · Zie FSA
 feature toggle · 76, 93
 federated identity management · 51
 feedback · 76, 77, 92, 93, 96, 99, 103
 feedforward · 93
 five times why methode · 14
 flow · 76, 77, 92, 95, 97, 98, 99, 100, 102, 103
 focus · 101
 framework · 98
 FSA · 106
 functie · 1
 Functionele AcceptatieTest · Zie FAT

G

GAT · 106
 Gaussian distribution · 89, 93
 GCC · 7, 106
 GDPR · 106
 Gebruiker AcceptatieTest · Zie GAT
 gebruikersorganisatie · 8
 Gene Kim · 23, 93, 98, 102
 General Computer Controls · Zie GCC
 General Data Protection Regulation · Zie GDPR
 Generieke & Specifieke Acceptatiecriteria · Zie GSA
 Geo-backup · 52
 gereedschap · 76
 gewenste situatie · 8
 Gherkin language · 27
 GIT · 106
 Given When Then · 93, Zie GWT
 Global Information Tracker · Zie GIT
 Graphical User Interface · Zie GUI

green build · 76
 green field · 94
 GSA · 106
 GSA aanpak · 64
 GUI · 106
 GWT · 93, 106

H

Hand-off Readiness Review · Zie HRR
 happy flow · 33
 happy path · 89, 94
 hardware · 92, 95, 103
 holocracy · 94
 horizontal splitting of feature · 94, 103
 HRM · 7, 25, 106
 HRR · 76, 106
 huidige situatie · 8
 Human Resource Management · Zie HRM
 hypothesis driven development · 76

I

IaC · 89, 95, 106
 IaC scripting · 55
 ICT · 95, 106
 ID · 106
 ideal test pyramid · 76, 102
 idempotent · 94
 Identifier · Zie ID
 imparative programming · 94
 Independent, Negotiable, Valuable,
 Estimatable, Small and Testable · Zie
 INVEST
 Information assets, People, Organisation,
 Products and services, Systems and
 processes · Zie IPOPS
 Information Communication Technology ·
 Zie ICT
 information radiator · 95
 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 · 95
 Infrastructure as Code · Zie IaC
 Infrastructure Risk · Zie IR
 infrastructuur
 - architectuur · 8
 - component · 95
 - management · 95
 integrity · 105
 International Standard On Assurance
 Engagements · Zie ISAE

INVEST · 94, 106
 IP address · 95
 IPOPS · 106
 IR · 106
 ISAE 3402 type 2 · 55
 I-shaped · 76, 94
 ISMS · 106
 ISO · 107
 ISO 27001 · 39
 IST · 96
 IST situatie · 8
 ISVS · 107
 IT · IV, 7, 93, 97, 102, 107
 ITIL 4 · 107
 ITSM · 97, 107

J

Java Virtual Machine · Zie JVM
 JIC · 107
 Ji-Kotei-Kanketsu · Zie, Zie JKK
 JIT · 95, 96, 107
 JKK · 95, 107
 Just In Case · Zie JIC
 Just In Time · Zie JIT
 JuVoTA · 44
 JVM · 107

K

Kaizen · 95, 97
 Kaizen Blitz (or Improvement Blitz) · 96
 Kaizen in advance · 96
 Kanban · 96, 97, 104
 keten · 76
 Key Performance Indicator · Zie KPI
 kibana dashboard · 96
 KPI · 96, 97, 104, 107
 Kritieke Succes Factor · Zie KSF
 KSF · 107
 kwaliteitscriteria · 77

L

LAN · 107
 latent defect · 96
 Launch Readiness Review · Zie LRR
 launching guidance · 96
 LCM · 107
 LDAP · 107
 Lead Time · 96, Zie LT
 Lean · 102, 103
 Lean indicator · 11
 Lean tool · 97
 learning culture · 97
 levenscyclus · 26, 27
 lifecycle · 93, 95
 LifeCycle Management · Zie LCM

Lightweight Directory Access Protocol · Zie LDAP
 lijnmanager · 1
 Local Area Network · Zie LAN
 log · 76, 98
 logging level · 97
 loosely coupled architecture · 97
 loosely coupled services · 97
 LRR · 76, 96, 107
 LT · 96, 107

M

Machine Learning · Zie ML
 machtsverhouding · 3, 17, 20, 21, 22
 management · 78
 Management Information · Zie MI
 manufacturing process · 103
 marker · 37
 MASR · 107
 Mean Time Between Failure · Zie MTBF
 Mean Time Between System Incidents · Zie MTBSI
 Mean Time To Repair · Zie MTTR
 merging · 76
 metadata · 90
 methodology · 78
 metrics · 76
 MFA · 107
 MI · 107
 microservice · 31, 97
 microservice architecture · 97
 Microsoft Operations Framework · Zie MOF
 mini pipeline · 97
 Minimal Viable Product · Zie MVP
 Minimum Required Information · Zie MRI
 MITM · 51
 ML · 107
 Modify, Avoid, Share, Retain · Zie MASR
 Module Test · Zie MT
 MOF · 107
 monitor architecture · 76
 monitoring · 98
 monitoringresultaten · 42
 monitorvoorziening · 22
 monolithic · 98
 MRI · 97, 107
 MT · 107
 MTBF · 107
 MTBSI · 107
 MTTR · 98, 107
 muda · 98
 Multi Factor Authentication · Zie MFA
 MVP · 10, 107

N

NC · 107
 NFR · 91, 98, 107
 Non Conformity · Zie NC
 Non Functional Requirement · Zie NFR

O

OAWOW · 107
 obeya · 98
 object code · 90
 OLA · 107
 One Agile Way of Working · Zie OAWOW
 one piece flow · 98
 ontwerp · 8
 ontwerpcriteria · 53
 Ontwikkel-, Test-, Acceptatie- en Productieomgeving · Zie OTAP
 Operational Level Agreement · Zie OLA
 operations · 89, 93, 98, 100, 101, 103
 operations story · 98
 Ops engineer · 1
 Ops liaison · 98
 organisatievormgeving · 3
 organisation archetype · 98
 organisational typology model · 98
 OTAP · 107
 OTAP-straat · 25, 38
 over-the-shoulder · 76, 99

P

PAAS · 50, 107
 package · 99
 pair-programming · 76, 91, 99
 PAT · 89, 107
 patroonherkenning · 42
 pattern · 90, 99
 Pattern

- Applicatie Service · 2
- Cloud Service · 2
- DevOps Service · 2
- Hybride Service · 2
- Specialisatie Service · 2

 PBI · 107
 PDCA · 96, 97, 107
 peer review · 99
 peer to peer programming · 91
 PEN-test · 60
 People, Process & Technology · Zie PPT
 performance · 91, 97, 103, 104, 107
 Performance StressTest · Zie PST
 performancecriterium · 35
 PESTLE · 107
 pipeline · 76, 77, 89, 95, 97, 100, 101, 104
 Plan Do Check Act · Zie PDCA
 planningsobject · 9, 10
 Platform As A Service · Zie PAAS
 Political, Economic, Sociological, Technological, Legislative, Environmental · Zie PESTLE
 POR · 107
 post mortem · 99
 PPT · 25, 107
 PPT-aspect · 76
 principe · 18
 probleem · 13, 14, 15

proceseigenaar · 1
 procesmanager · 1
 Processing Time · Zie PT
 product

- backlog · 19, 94, 95, 99
- backlog item · 94
- owner · 1, 99

 Product Backlog Item · Zie PBI
 Productie AcceptatieTest · Zie PAT
 production environment · 97
 programming paradigm · 99
 Project or Organisational Risk · Zie POR
 PSQL · 92
 PST · 107
 PT · 107
 pull request process · 76, 99

Q

QA · 77, 99, 107
 QC · 107
 quality · 78
 Quality Assurance · Zie QA
 Quality Control · Zie QC

R

RACI · 108
 RASCI · 20, 21, 108
 RBAC · 50, 108
 Real User Monitoring · Zie RUM
 reduce batch size · 99
 reduce number of handoffs · 99
 redundantie · 19
 regievoerder · 21
 Reinforcement Learning · Zie RL
 release · 2, 7, 99
 release manager · 99
 release pattern · 99
 repository · 76, 77, 90, 91, 99, 100
 REpresentational State Transfer Application
 Programming Interface · Zie REST API
 requirement · 8, 76, 90, 96, 98, 101, 107
 research · 1
 resource · 3, 18
 Responsibility, Accountable, Consulted and
 Informed · Zie RACI
 Responsibility, Accountable, Supporting,
 Consulted and Informed · Zie RASCI
 REST API · 60, 108
 retrospective · 93
 Return On Investment · Zie ROI
 review · 93
 right to audit · 43
 risico · 1, 77, 90, 96, 127
 RL · 108
 roadmap · 17, 27, 45
 Robotic Process Automation · Zie RPA
 ROI · 108
 rol · 33, 34
 Role-Based Access Control · Zie RBAC

rollback technique · 76
 rootcause · 13
 rootcause analyse · 97
 RPA · 108
 RUM · 108

S

SA · 108
 SAAS · 50
 SAFe · 108
 safety check · 99
 Sarbanes Oxley · Zie SoX
 SAT · 108
 SBAR · 100, 108
 SBB · 35, 108
 SBB-A · 108
 SBB-I · 108
 SBB-T · 35, 108
 Scaled Agile Framework · Zie SAFe
 S-CI · 108
 Scrum master · 1
 Secure Sockets Layer · Zie SSL
 security · 91, 98, 100, 102
 Security Acceptatie Test · Zie SAT
 security auditing · 45
 security officer · 91
 Security, Risk of Compliancy · Zie SRC
 self service capability · 100
 service · 107
 Service Integration And Management · See
 SIAM
 Service Level Agreement · Zie SLA
 service providers · 22
 Service Value System · Zie SVS
 shared goals · 100
 silo · 103
 Simian army · 76, 100, 102
 Simple Network Management Protocol · Zie
 SNMP
 SIT · 108
 Situation, Background, Assessment,
 Recommendation · Zie SBAR
 skills · 25, 94
 SLA · 21, 77, 86, 108
 SLA breaches · 49
 SLA control · 41
 SM · 108
 SMART · 97, 108
 SME · 108
 smoke testing · 100
 SNMP · 108
 SoA · 108
 SoE · 7, 101, 108
 SOE · 31
 software · 78, 90, 102, 103, 104
 Software Configuration Item · Zie S-CI
 softwareontwikkelproces · 78
 SoI · 7, 101, 108
 SOI · 31
 SOLL · 96
 SOLL situatie · 8

SoR · 7, 101, 108
 SOR · 31
 sourcecode · 76, 90, 91, 93, 100, 101, 103
 SoX · 108
 SoX-wetgeving · 7
 Specific, Measurable, Accountable, Realistic, Timely · Zie SMART
 sprint · 10, 93
 sprint execution · 93
 sprint planning · 93
 SQL · 108
 SQP · 80
 SRC · 108
 SRG · 76, 77, 108
 SSL · 108
 ST · 108
 stakeholder · 9, 10, 25, 37, 93, 94
 standard deviation · 100
 standard operations · 101
 Standard Rules & Guidelines · Zie SRG
 stand-up · 93
 Statement of Applicability · Zie SoA
 static analysis · 101
 statusbewaking · 10
 story · 10
 Strategic IS Architecture · Zie SA
 Strategic Match · Zie SM
 strategie · 8
 strategy · 78
 Strength, Weakness, Opportunities, Threats · Zie SWOT
 Structured Query Language · Zie SQL
 Subject Matter Expert · Zie SME
 super use case · 35
 SVS · 108
 SWOT · 108
 System Building Block · Zie SBB
 System Building Block Application · Zie SBB-A
 System Building Block Infrastructure · Zie SBB-I
 System Building Block Technology · Zie SBB-T
 System Building Blocks · 36
 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 · 89
 Taken, Verantwoordelijkheden en Bevoegdheden · Zie TVB
 Target Operating Model · See TOM, Zie TOM
 task · 10, 95
 TCO · 28, 108
 TCP · 108
 TDD · 76, 101, 108
 Team Foundation Server · Zie TFS

technical debt · 17, 78, 92, 93
 technical debt backlog · 17, 78
 Technical Information Security Officer · Zie TISO
 Technical Uncertainty · Zie TU
 technology adaption curve · 101
 technology executive · 101
 telemetry · 76
 template · 30, 35, 75
 test
 - case · 89, 90, 91
 - harness · 102
 - management · 76
 Test Driven Development · Zie TDD
 testbasis · 52
 tester · 93
 TFS · 108
 The Agile Manifesto · 102
 the ideal testing automation pyramid · 102
 The Lean movement · 102
 the non-ideal testing automation inverted pyramid · 102
 The Three Ways · 98, 102
 theme · 9, 10, 77
 theory of constraints · 103
 Time To Market · Zie TTM
 TISO · 108
 TOM · 108
 tool-assisted code review · 76, 103
 Total Cost of Ownership · Zie TCO
 Toyota Kata · 103
 Toyota Production System · Zie TPS
 TPA · 49
 TPM · 49
 TPS · 11, 108
 traceability · 76
 traceerbaarheid · 76, 77
 trage feedback · 13
 transformation team · 103
 Transmission Control Protocol · Zie TCP
 trunk · 100
 T-shaped · 76, 94
 TSQL · 92
 TTM · 108
 TU · 109
 TVB · 109

U

UAT · 109
 UML · 109
 Unified Modeling Language · Zie UML
 Unit Test · Zie UT
 Use Case · 33, 34, 35, 36, 39
 Use Case Diagram · 33, 34, 35, 36
 User Acceptance Test · Zie UAT
 User eXperience design · Zie UX design
 UT · 109
 UX design · 76, 109

V

value chain · 10, 11
value stream · 10, 11, 18, 27, 33, 34, 36, 76, 77, 93, 97, 100, 101, 103
Value Stream Canvas · 36
Value Stream Mapping · 36, Zie VSM
VCR · 109
velocity · 90
veranderparadigma · 3, 17, 18, 20, 21, 22
verantwoordelijkheden · 3, 17
versiebeheer · 76
vertical splitting of feature · 103
Verwachte Contante ROI · Zie VCR
virtualised environment · 103
visibility · 76
visie · 18
visualisatie · 11, 103
Voice Over Internet Protocol · Zie VOIP
VOIP · 109
volwassenheid · 80
volwassenheidsniveau · 78
VSM · 77, 97, 103, 109
vulnerability scan · 53

W

walking skeleton · 103

WAN · 109
war room · 98
waste · 90, 92, 95, 97, 98, 102, 103, 104, 137
waste reductie · 104
Way of Working · Zie WoW
Westrum · 98, 99
Wide Area Network · Zie WAN
Windows Management Instrumentation · Zie WMI
WIP · 109
WMI · 109
Work In Progress · Zie WIP
work item · 10
workflow · 92
WoW · 18, 109

X

XML · 109
XP · 109

Z

Zachman · 8

Nawoord

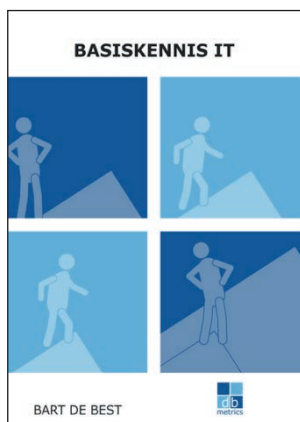
Mijn ervaring is dat de denkbeelden die ik vastleg in een artikel of een boek zich blijven evolveren. In geval u met een bepaald onderwerp uit dit boek aan de slag gaat in uw eigen DevOps organisatie, dan raad ik u aan om even met mij contact op te nemen. Wellicht zijn er aanvullende artikelen of ervaringen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

Over de auteur



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

Andere boeken van deze auteur



Basiskennis IT

De eerste stap van een leven lang leren.

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

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

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

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



SLA Best Practices

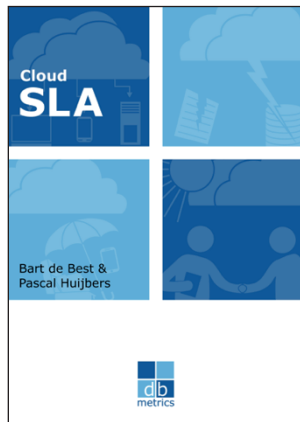
Het volledige ABC van service level agreements.

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

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

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

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



Cloud SLA

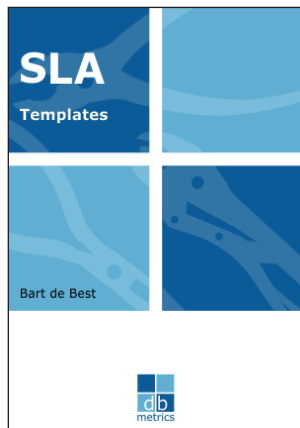
De best practices van cloud service level agreements

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

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

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

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



SLA Templates

A complete set of SLA templates

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

The following templates are included in this book:

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

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



ICT Prestatie-indicatoren

De beheerorganisatie meetbaar gemaakt.

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

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

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



Quality Control & Assurance

Kwaliteit op maat.

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

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

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

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



Acceptatiecriteria

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

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

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

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



Beheren onder Architectuur

Het richting geven aan de inrichting van beheerorganisaties.

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

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

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



Agile Service Management met Scrum

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

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

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

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



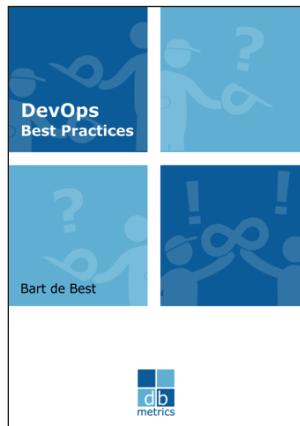
Agile Service Management met Scrum in de Praktijk

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

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

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

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



DevOps Best Practices

Best Practices for DevOps

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

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

Author : Bart de Best
 Publisher : Leonon Media, 2017 (UK), 2018 (Pocket Guide)
 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 to 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 Outcome

Een uitgave in de Continuous Everything reeks.

Om BizDevOps echt in te vullen moeten de werelden van business en DevOps hechter samenwerken. Dat kan alleen als zij dezelfde taal spreken. Continuous Outcome geeft hier invulling aan door de business value streams te beschouwen vanuit een Continuous Everything perspectief. Dit is gedaan door het DevOps Lemniscaat te gebruiken om de stappen van een business value stream te beschrijven. Natuurlijk zijn de business value streams uniek per organisatie, maar op een abstractieniveau zijn het ook value streams die ontwikkelen en beheren.

Dit boek is een publicatie in de continuous everything reeks. De content bestaat uit het afbeelden van Continuous Everything best practices op de werkwijzen van de business.

Na de definitie van de Continuous Outcome value stream wordt voor elke use case uit die value stream gekeken welke Continuous Everything best practices toegepast kunnen worden en wat het belang is om hierin samen te werken. Met deze holistische benadering van Continuous Outcome is een optimale en integrale invulling mogelijk van BizDevOps.

Auteur : Bart de Best
 Uitgever : Leonon Media, 2024
 ISBN (NL) : 978 94 91480 386
 ISBN (UK) : 978 94 91480 393



Continuous Architecture

Een uitgave in de Continuous Everything reeks.

Continuous Architecture richt zich op het borgen dat de organisatiestrategie wordt gerealiseerd door richting te geven aan de innovatie en beheer van de informatievoorziening die daartoe nodig is. Hierbij wordt een onderscheid gemaakt tussen System of Records (ketenapplicaties), System of Engagement (enkelvoudige applicaties) en System of Services (servicearchitectuur voor ontwikkeling en beheer)

Dit boek is een publicatie in de continuous everything reeks. De content bestaat uit de bespreking van de value streams voor de realisatie van de SoR, SoE en SOS systems.

Tevens bevat het voorbeeld architecture principes en modellen per Continuous Everything value stream. Met deze holistische benadering van Continuous Architecture is een optimale en integrale invulling mogelijk voor zowel de ontwikkeling als beheer van de informatievoorziening van de gehele organisatie die nodig is voor de realisatie van de organisatiestrategie.

Auteur : Bart de Best
 Uitgever : Leonon Media, 2024
 ISBN (NL) : 978 94 91480 348
 ISBN (UK) : 978 94 91480 355



Continuous Planning

Een uitgave in de Continuous Everything reeks.

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

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

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



Continuous Design

Een uitgave in de Continuous Everything reeks.

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

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

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



Continuous SLA

Een uitgave in de Continuous Everything reeks.

Continuous SLA richt zich op het onderkennen van risico's die de outcome van bedrijfsprocessen (core value streams) kunnen schaden. Deze risico's ontstaan als gevolg van nieuwbouw en onderhoud van informatiesystemen middels Agile teams. Binnen het concept van Continuous SLA worden deze risico's vanuit verschillende perspectieven geanalyseerd en door het DevOps team voorzien van tegenmaatregelen ook wel SLA controls genoemd. Door deze SLA controls meetbaar te maken worden het geschikte planningsobjecten die op de productbacklog kunnen worden gezet.

Dit boek is een publicatie in de continuous everything reeks. De content bestaat uit de bespreking van technieken om risico's te

onderkennen en beheersen zoals het gebruik van Lean indicatoren, value stream mappingen informatie-, applicatie- en technische architectuurbouwen. Naast de core value streams worden hiertoe ook de enable value streams zoals beheer, informatiebeveiliging en ontwikkel value streams onderzocht op risico's die direct of indirect de outcome schaden. De onderkende SLA controls worden verankerd in de Agile way of working door de samenwerking tussen onder andere de product owner en service level manager uit te diepen. Met deze integrale aanpak van SLA controls wordt het mogelijk om grip te krijgen op kwaliteit in Agile projecten.

Auteur : Bart de Best
 Uitgever : Leonon Media, 2023
 ISBN (NL) : 978 94 91480 263
 ISBN (UK) : 978 94 91480 256



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 Acceptance

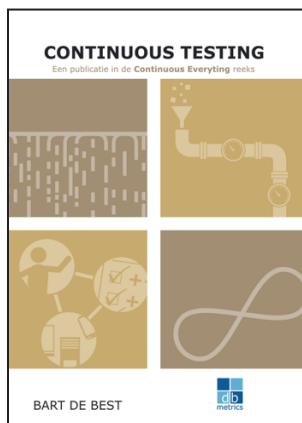
Een uitgave in de Continuous Everything reeks.

Continuous Acceptance richt zich op het accepteren van nieuwe en aangepaste producten en services die in een Agile omgeving zijn voortgebracht.

In deze CE value stream worden de specifieke acceptatiecriteria ontleend aan de business value streams door op zoek te gaan naar de risico's dat de doelen van de business niet gehaald worden. De tegenmaatregelen van deze risico's worden met acceptatietesten getoetst op effectiviteit. In analogie hierop worden de generieke acceptatiecriteria afgeleid uit de CE value streams die invulling geven aan het DevOps Lemniscaat.

Dit boek is een publicatie in de continuous everything reeks. De content bestaat uit de bespreking van de afleiding van acceptatiecriteria. Tevens wordt een voorbeeld uitwerking gegeven voor specifieke acceptatiecriteria en wordt er een aantal generieke acceptatiecriteria gegeven voor de volgende value streams: Continuous Planning, Continuous Design, Continuous Testing, Continuous Integration, Continuous AI, Continuous Deployment, Continuous Monitoring, Continuous Learning, Continuous Security, Continuous Auditing, Continuous SLA en Continuous Assessment. Hiermee heeft u een gereedschap in handen om grip te krijgen op de acceptatie van applicaties en services.

Auteur : Bart de Best
 Uitgever : Leonon Media, 2023
 ISBN (NL) : 978 94 91480 317
 ISBN (UK) : 978 94 91480 324



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 wor-

den 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 AI

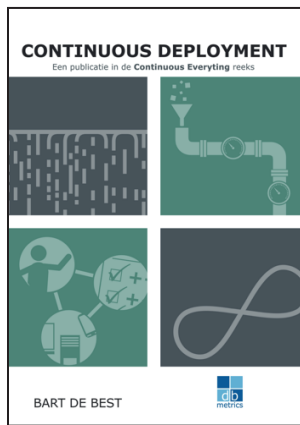
Een uitgave in de Continuous Everything reeks.

Continuous AI richt zich op het verhogen van de outcome van CE-aspectgebieden zoals de value streams Continuous Testing en Continuous Integration.

Van deze value streams worden de bottlenecks gelokaliseerd in de vorm van limitations (performance) en boundaries (functionaliteit). Deze bottlenecks kunnen middels AI-toepassingsgebieden verkleind of verwijderd worden zoals door de inzet van Machine Learning (ML) en Natural Language Processing (NLP). Dit boek is een publicatie in de continuous everything reeks. De content bestaat uit de bespreking van AI-toepassingsgebieden voor alle gepubliceerde CE-aspectgebieden.

Per stap uit elke value stream wordt aangegeven wat nu of in de toekomst de mogelijkheden zijn van AI. De betrokken value streams zijn: Continuous Planning, Continuous Design, Continuous Testing, Continuous Integration, Continuous Deployment, Continuous Monitoring, Continuous Learning, Continuous Security, Continuous Auditing, Continuous SLA en Continuous Assessment. Hiermee heeft u een gereedschap in handen om AI op een gestructureerde en effectieve manier in uw organisatie toe te passen.

Auteur : Bart de Best
 Uitgever : Leonon Media, 2023
 ISBN (NL) : 978 94 91480 294
 ISBN (UK) : 978 94 91480 300



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

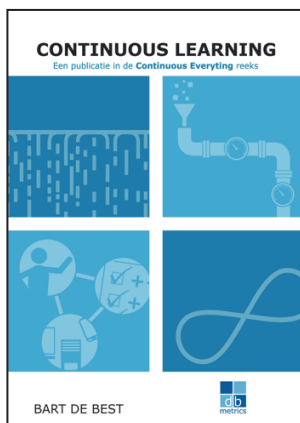
Een uitgave in de Continuous Everything reeks.

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 organisatiebehoeften 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 Service Integration

Een uitgave in de Continuous Everything reeks.

Continuous service integration omvat alle aspecten van BizDevSecOps om een geïntegreerde service aan de business te leveren door de samenwerking van meer service providers als ware het een single service provider service. Deze publicatie is feitelijk de uitleg van de samenwerking van de 15 overige continuous everything value streams, zoals deze zijn gedefinieerd in de continuous everything reeks. Zie ook de BizDevSecOps.Co Lemniscaat op www.bizdevsecops.co over deze value streams en wat deze omvatten.

Dit boek is een publicatie in de continuous everything reeks. De content bestaat uit de bespreking van 5 onderkende patterns van

geïntegreerde services en hoe de risico's van deze 5 onderkende patterns beheerst kunnen worden door de inzet van de value streams, zoals gedefinieerd in de continuous everything reeks. Voor elke pattern is een hoofdstuk geschreven met de definitie, de risico's en de tegenmaatregelen. Hierbij is steeds de relatie met de 15 gerelateerde continuous everything value streams gelegd waarin de borging van de risico's is gedefinieerd.

Author : Bart de Best
 Publisher : Leonon Media, 2024
 ISBN (NL) : 978 94 91480 423
 ISBN (UK) : 978 94 91480 430



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 Business

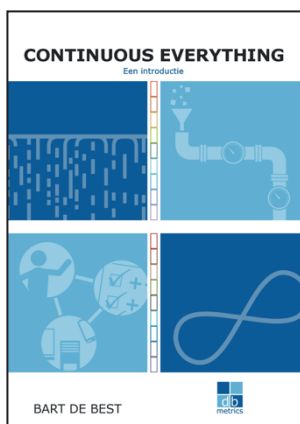
Een uitgave in de Continuous Everything reeks.

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 vier Continuous Everything boeken te weten: Continuous Outcome, Continuous Architecture, Continuous Acceptance en Continuous AI. 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 BizDevOps.

Author : Bart de Best
 Publisher : Leonon Media, 2024
 ISBN (NL) : 978 94 91480 362
 ISBN (UK) : 978 94 91480 379



Continuous Everything een Introductie

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 introductie van Continuous Everything aspecten te weten: Continuous Planning, Continuous Design, Continuous Testing, Continuous Integration, Continuous Deployment, Continuous Monitoring en Continuous Learning. Voor elk Continuous Everything aspectgebied wordt behandeld wat de basisconcepten zijn, wat de definitie is, wat de op te lossen problemen zijn en worden de belangrijkste modellen besproken. Met dit boek in de hand heeft u goed overzicht wat Continuous Everything inhoudt.

Author : Bart de Best
 Publisher : Leonon Media, 2023
 ISBN (NL) : 978 94 91480 270



Continuous Everything

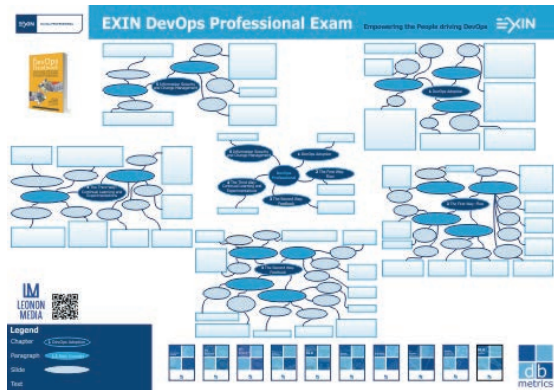
Een uitgave in de Continuous Everything reeks.

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.

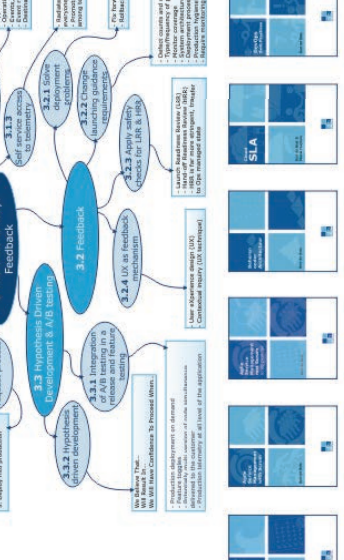
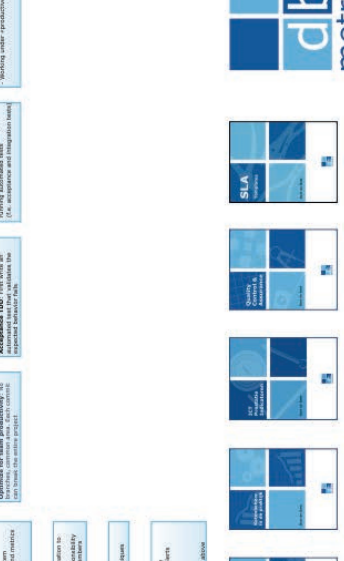
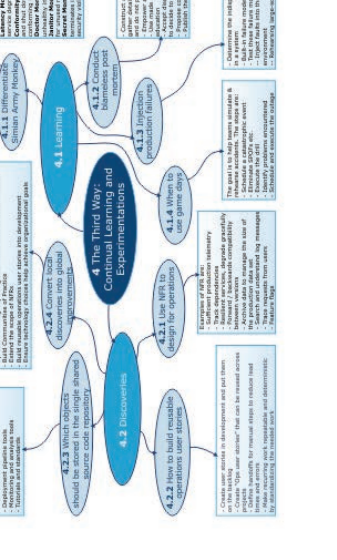
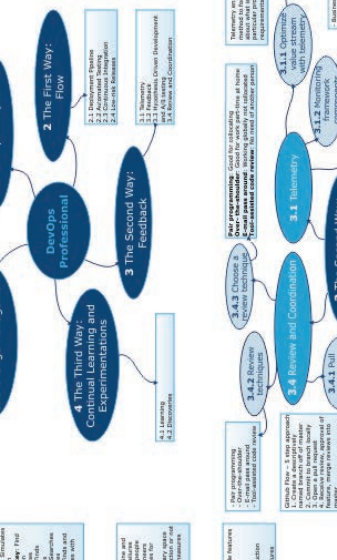
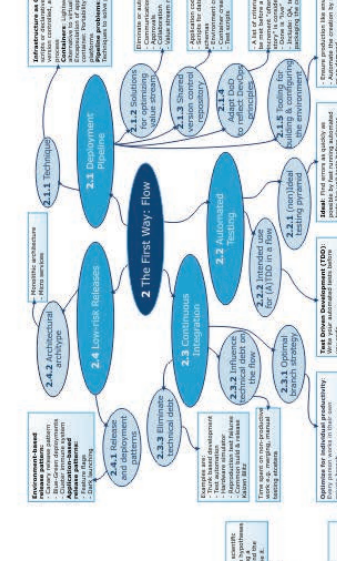
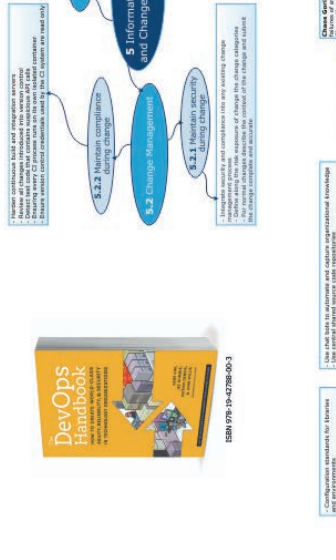
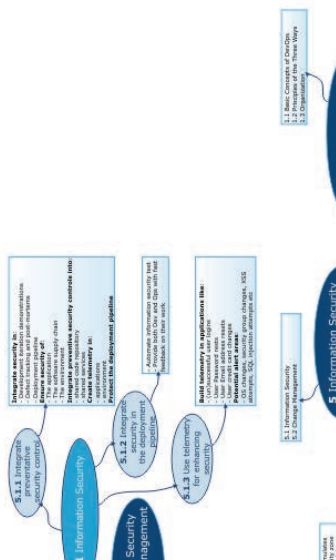
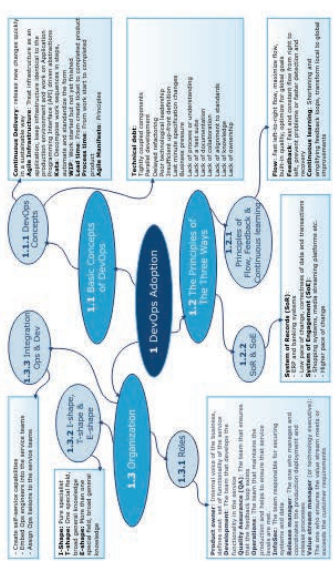
The subjects on the poster are based on the basic training material of Exin. Since there are many terms to be learned, this poster will help to learn them by reviewing them all at once daily.

Author : Bart de Best
 Publisher : Leonon Media, 2018
 Ordering : info@leonon.nl

EXIN DevOps Professional Exam

Empowering the People driving DevOps

EXIN DevOps PROFESSIONAL



Legend

Chapter

Paragraph

Slide

Text

Abbreviations:

- CI Continuous Integration
- DevOps Development as Code
- IaC Infrastructure as Code
- SaaS Software as a Service
- SRE Site Reliability Engineering
- Ops Operations
- SaaS System of Engagement
- SaaS System of Integration
- UX User Experience
- WSP Weighted Shortest Job First

EXIN DevOps PROFESSIONAL



CONTINUOUS SERVICE INTEGRATION

Een publicatie in de
Continuous Everything
reeks

Bart de Best

Continuous service integration omvat alle aspecten van BizDevSecOps om een geïntegreerde service aan de business te leveren door de samenwerking van meer service providers als ware het een single service provider service. Deze publicatie is feitelijk de uitleg van de samenwerking van de 15 overige continuous everything value streams, zoals deze zijn gedefinieerd in de continuous everything reeks. Zie ook de BizDevSecOps.Co Lemniscaat op www.bizdevsecops.co over deze value streams en wat deze omvatten.

Dit boek is een publicatie in de continuous everything reeks. De content bestaat uit de bespreking van 5 onderkende patterns van geïntegreerde services en hoe de risico's van deze 5 onderkende patterns beheerst kunnen worden door de inzet van de value streams, zoals gedefinieerd in de continuous everything reeks. Voor elke pattern is een hoofdstuk geschreven met de definitie, de risico's en de tegenmaatregelen. Hierbij is voor elke pattern de relatie met de 15 gerelateerde continuous everything value streams uitgelegd waarin de borging van de risico's is gedefinieerd.

