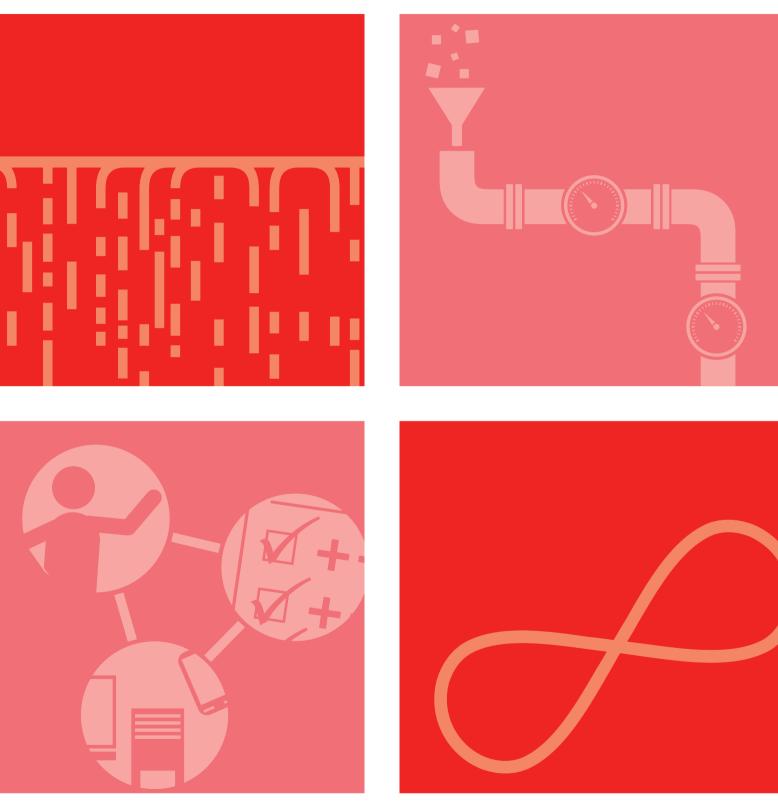
CONTINUOUS ARCHITECTURE

A publication in the **Continuous Everyting** series







DevOps Continuous Architecture Best Practices

A publication in the Continuous Everything series

Bart de Best

Edited by Louis van Hemmen

Colophon

More information about this and other publications can be obtained from: Leonon Media (0)572 - 851 104

Common questions Sales questions Manuscript / Author	: info@leonon.nl : verkoop@leonon.nl : redactie@leonon.nl
© 2024 Leonon Media	
Cover design Production	: Eric Coenders, IanusWeb, Nijmegen : Printforce B.V., Culemborg
Title	: DevOps Continuous Architecture
Subtitle	: A publication in the Continuous Everything series
Date	: 2 February 2024
Author	· Bart do Bost

Inte	. Devops continuous Architecture
Subtitle	: A publication in the Continuous Everything series
Date	: 2 February 2024
Author	: Bart de Best
Publisher	: Leonon Media
ISBN13	: 978 94 91480 355
Edition	: First press, first edition, 2 February 2024

© 2024, Leonon Media

No part of this publication may be reproduced and/or published by means of print, photocopy, microfilm, or any other means without the prior written consent of the publisher.

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

Table of Contents

1.1 OBJECTIVE 1 1.2 TARGET GROUP 1 1.3 BACKGROUND 1 1.4 STRUCTURE 2 1.5 APPENDIX 3 1.6 READING GUIDELINES 3 2 BASIC CONCEPTS EN BASIC TERMS 5 2.1 BASIC CONCEPTS EN BASIC TERMS 5 2.2 BASIC CONCEPTS 5 2.3 BASIC CONCEPTS 7 3 CONTINUOUS ARCHITECTURE DEFINITION 11 3.1 BACKGROUND 11 3.2 IMPLEMENTATION 11 3.2 IMPLEMENTATION 11 3.2 INPLEMENTATION 11 3.2 INPLEMENTATION 11 3.4 ORGANISATION 15 4.1 THE CHANGE PARADIGM 16 4.3 POWER 17 4.4 ORGANISATION 20 5.1 ARCHITECTURE RACHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PUNCIPLES 23 5.2 ARCHITECTURE VALUE STREAM 31 <tr< th=""><th>1</th><th>INTRODUCTION</th><th> 1</th></tr<>	1	INTRODUCTION	1
1.2 TARGET GROUP 1 1.3 BACKGROUND 1 1.4 STRUCTURE 2 1.5 APPENDIX 3 1.6 READING GUIDELINES 3 2 BASIC CONCEPTS EN BASIC TERMS 5 2.1 BASIC CONCEPTS EN BASIC TERMS 5 2.2 BASIC CONCEPTS 5 2.3 BASIC CONCEPTS EN BASIC TERMS 7 3 CONTINUOUS ARCHITECTURE DEFINITION 11 3.1 BACKGROUND 11 3.1 CONTINUOUS ARCHITECTURE ANCHORING 15 4.1 THE CHANGE PARADIGM 15 4.2 VISION 16 13 4.3 POWER 17 4.4 4.5 RESOURCES 20 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 </th <th></th> <th></th> <th></th>			
1.3 BACKGROUND 1 1.4 STRUCTURE 2 1.5 APPENDIX 3 1.6 READING GUIDELINES 3 2 BASIC CONCEPTS EN BASIC TERMS 5 2.1 BASIC CONCEPTS 5 2.2 BASIC TERMS 7 3 CONTINUOUS ARCHITECTURE DEFINITION 11 3.1 BACKGROUND 11 3.2 IMPLEMENTATION 11 3.1 BACKGROUND 11 3.2 IMPLEMENTATION 11 3.2 IMPLEMENTATION 11 3.4 CONTINUOUS ARCHITECTURE ANCHORING 15 4.1 THE CHANGE PARADIGM 15 4.2 VISION 16 4.3 POWER 17 4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.2 ARCHITECTURE PINICIPLES 23 5.2 ARCHITECTURE PINICIPLES 23 5.2 ARCHITECTURE VALUE STREAM 31 <			
1.4 STRUCTURE 2 1.5 APPENDIX 3 1.6 READING GUIDEUNES 3 2 BASIC CONCEPTS EN BASIC TERMS 5 2.1 BASIC CONCEPTS 5 2.2 BASIC CONCEPTS 5 2.2 BASIC CONCEPTS 5 2.2 BASIC CONCEPTS 5 2.2 BASIC CONCEPTS 5 2.4 BASIC CONCEPTS 5 2.5 BASIC CONCEPTS 7 3 CONTINUOUS ARCHITECTURE DEFINITION 11 3.1 BACKGROUND 11 3.2 IMPLEMENTATION 11 4.1 THE CHANGE PARADIGM 15 4.2 VISION 16 4.3 POWER 17 4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE VALUE STREAM 31 6.1 CONTINUOUS ARCHITECTURE VALUE STREAM 31 <	1.3		
1.6 READING GUIDELINES 3 2 BASIC CONCEPTS EN BASIC TERMS 5 2.1 BASIC CONCEPTS 5 2.2 BASIC TERMS 7 3 CONTINUOUS ARCHITECTURE DEFINITION 11 3.1 BACKGROUND 11 3.2 IMPLEMENTATION 11 3.4 CONTINUOUS ARCHITECTURE ANCHORING 15 4.1 THE CHANGE PARADIGM 15 4.2 VISION 16 4.3 POWER 17 4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE VALUE STREAM 31 6.1 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.2 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CON	1.4		
1.6 READING GUIDELINES 3 2 BASIC CONCEPTS EN BASIC TERMS 5 2.1 BASIC CONCEPTS 5 2.2 BASIC TERMS 7 3 CONTINUOUS ARCHITECTURE DEFINITION 11 3.1 BACKGROUND 11 3.2 IMPLEMENTATION 11 3.4 CONTINUOUS ARCHITECTURE ANCHORING 15 4.1 THE CHANGE PARADIGM 15 4.2 VISION 16 4.3 POWER 17 4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE VALUE STREAM 31 6.1 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.2 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CON	1.5		
2.1 BASIC CONCEPTS 5 2.2 BASIC TERMS 7 3 CONTINUOUS ARCHITECTURE DEFINITION 11 3.1 BACKGROUND 11 3.2 IMPLEMENTATION 11 3.4 CONTINUOUS ARCHITECTURE ANCHORING 15 4.1 THE CHANGE PARADIGM 15 4.2 VISION 16 4.3 POWER 17 4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE MODELS 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE DESIGN 31 6.2 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.4 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37			
2.1 BASIC CONCEPTS 5 2.2 BASIC TERMS 7 3 CONTINUOUS ARCHITECTURE DEFINITION 11 3.1 BACKGROUND 11 3.2 IMPLEMENTATION 11 3.4 CONTINUOUS ARCHITECTURE ANCHORING 15 4.1 THE CHANGE PARADIGM 15 4.2 VISION 16 4.3 POWER 17 4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE MODELS 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE DESIGN 31 6.2 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.4 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37	2		-
2.2 BASIC TERMS 7 3 CONTINUOUS ARCHITECTURE DEFINITION 11 3.1 BACKGROUND 11 3.2 IMPLEMENTATION 11 4 CONTINUOUS ARCHITECTURE ANCHORING 15 4.1 THE CHANGE PARADIGM 15 4.2 VISION 16 4.3 POWER 17 4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE MODELS 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37			
3 CONTINUOUS ARCHITECTURE DEFINITION			
3.1 BACKGROUND 11 3.2 IMPLEMENTATION 11 3.2 IMPLEMENTATION 11 4 CONTINUOUS ARCHITECTURE ANCHORING 15 4.1 THE CHANGE PARADIGM 15 4.2 VISION 16 4.3 POWER 17 4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE MODELS 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.2 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE 35 6.4 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE 35 6.5 CONTINUOUS ARCHITECTURE 41 7.1 THE STEPS 41	2.2	BASIC TERMS	/
3.2 IMPLEMENTATION 11 4 CONTINUOUS ARCHITECTURE ANCHORING 15 4.1 THE CHANGE PARADIGM 15 4.2 VISION 16 4.3 POWER 17 4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE MODELS 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE VALUE STREAM 31 6.2 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE 41 7.1 THE STEPS 41 7.1 THE STEPS 41 7.2 STEP 1.1 DETERMINE BUSINESS ACALS AND STRATEGY 55 <th>3</th> <th>CONTINUOUS ARCHITECTURE DEFINITION</th> <th>11</th>	3	CONTINUOUS ARCHITECTURE DEFINITION	11
4 CONTINUOUS ARCHITECTURE ANCHORING 15 4.1 THE CHANGE PARADIGM 15 4.2 VISION 16 4.3 POWER 17 4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE MODELS 23 5.2 ARCHITECTURE MODELS 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE USE REAM 31 6.2 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE 41 7.1 THE STEPS 41 7.2 STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY 41 7.3 STEP 1.2 DETERMINE BUSINESS GOALS AND STRATEGY 41 7.4 STEP 1.3 DETERMINE INFRASTRUCTURE ARCHITECTURE 48 7.5 STEP 1.4 DETERMINE APPLICATION ARCHITECTURE 56 7.4 S	3.1	BACKGROUND	11
4.1 THE CHANGE PARADIGM 15 4.2 VISION 16 4.3 POWER 17 4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE MODELS 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE VALUE STREAM 31 6.2 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE 35 6.4 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE 41 7.3 STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY 41 7.3 STEP 1.2 DETERMINE BUSINESS ARCHITECTURE 44 7.4 STEP 1.3 DETERMINE BUSINESS ARCHITECTURE 48 7.5 STEP 1.4 DETERMINE BUSINESS ARCHITECTURE 48 7.5 STEP 1.4 DETERMINE BUSINESS ARCHITECTURE 52	3.2	IMPLEMENTATION	11
4.1 THE CHANGE PARADIGM 15 4.2 VISION 16 4.3 POWER 17 4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE MODELS 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE VALUE STREAM 31 6.2 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE 35 6.4 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE 41 7.3 STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY 41 7.3 STEP 1.2 DETERMINE BUSINESS ARCHITECTURE 44 7.4 STEP 1.3 DETERMINE BUSINESS ARCHITECTURE 48 7.5 STEP 1.4 DETERMINE BUSINESS ARCHITECTURE 48 7.5 STEP 1.4 DETERMINE BUSINESS ARCHITECTURE 52	Л		15
4.2 VISION 16 4.3 POWER. 17 4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE MODELS 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE VALUE STREAM 31 6.2 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE 35 6.4 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE 41 7.1 THE STEPS 41 7.2 STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY 41 7.3 STEP 1.2 DETERMINE BUSINESS ARCHITECTURE 44 7.4 STEP 1.3 DETERMINE BUSINESS ARCHITECTURE 44 7.4 STEP 1.4 DETERMINE APPLICATION ARCHITECTURE 50 7.6 STEP 1.4 DETERMINE APPLICATION ARCHITECTURE 50 <			
4.3 POWER 17 4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE MODELS 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE DESIGN 31 6.2 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE 35 6.4 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE 41 7.1 THE STEPS 41 7.2 STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY 41 7.3 STEP 1.2 DETERMINE BUSINESS ACHITECTURE 44 7.4 STEP 1.2 DETERMINE BUSINESS ACHITECTURE 48 7.5 STEP 1.4 DETERMINE BUSINESS ACHITECTURE 50 7.6 STEP 1.4 DETERMINE APPLICATION ARCHITECTURE 50 7.6 STEP 1.4 DETERMINE APPLICATION ARCHITECTURE 50 7.6 STEP 1.5 DETERMINE APPLICATION ARCHITECTURE <t< td=""><td></td><td></td><td></td></t<>			
4.4 ORGANISATION 20 4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE MODELS 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE VALUE STREAM 31 6.2 CONTINUOUS ARCHITECTURE VALUE STREAM 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE 35 6.4 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE 41 7.1 THE STEPS 41 7.2 STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY 41 7.3 STEP 1.2 DETERMINE BUSINESS ARCHITECTURE 44 7.4 STEP 1.2 DETERMINE INFORMATION ARCHITECTURE 50 7.6 STEP 1.4 DETERMINE APPLICATION ARCHITECTURE 50 7.6 STEP 1.5 DETERMINE APPLICATION ARCHITECTURE 52 7.7 STEP 1.6 DETERMINE APPLICATION ARCHITECTURE 52 7.6 STEP 1.7 DE			
4.5 RESOURCES 21 5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES 23 5.2 ARCHITECTURE MODELS 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE DESIGN 31 6.2 CONTINUOUS ARCHITECTURE VALUE STREAM 31 6.2 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE 35 6.4 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE 41 7.1 THE STEPS 41 7.2 STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY 41 7.3 STEP 1.2 DETERMINE BUSINESS ARCHITECTURE 44 7.4 STEP 1.3 DETERMINE BUSINESS ARCHITECTURE 42 7.5 STEP 1.4 DETERMINE APPLICATION ARCHITECTURE 50 7.6 STEP 1.5 DETERMINE HIGH APPLICATION ARCHITECTURE 52 7.7 STEP 1.6 DETERMINE APPLICATION ARCHITECTURE 52 7.8<			
5 CONTINUOUS ARCHITECTURE ARCHITECTURE 23 5.1 ARCHITECTURE PRINCIPLES. 23 5.2 ARCHITECTURE MODELS. 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE DESIGN 31 6.2 CONTINUOUS ARCHITECTURE VALUE STREAM 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE 35 6.4 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE 41 7.1 THE STEPS 41 7.2 STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY 41 7.3 STEP 1.2 DETERMINE BUSINESS ARCHITECTURE 44 7.4 STEP 1.3 DETERMINE BUSINESS ARCHITECTURE 44 7.4 STEP 1.4 DETERMINE APPLICATION ARCHITECTURE 50 7.6 STEP 1.5 DETERMINE APPLICATION ARCHITECTURE 52 7.7 STEP 1.6 DETERMINE APPLICATION ARCHITECTURE 52 7.7 STEP 1.5 DETERMINE APPLICATION ARCHITECTURE 56 7.8 STEP 1.7 DETERMINE BUSINESE CAADMAP 56			
5.1 ARCHITECTURE PRINCIPLES. 23 5.2 ARCHITECTURE MODELS. 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE VALUE STREAM 31 6.2 CONTINUOUS ARCHITECTURE VALUE STREAM 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE 35 6.4 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE 41 7.1 THE STEPS 41 7.2 STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY. 41 7.3 STEP 1.2 DETERMINE BUSINESS GOALS AND STRATEGY. 41 7.3 STEP 1.2 DETERMINE BUSINESS ARCHITECTURE 44 7.4 STEP 1.3 DETERMINE INFORMATION ARCHITECTURE 48 7.5 STEP 1.4 DETERMINE APPLICATION ARCHITECTURE 50 7.6 STEP 1.5 DETERMINE INFRASTRUCTURE ARCHITECTURE 52 7.7 STEP 1.5 DETERMINE INFRASTRUCTURE ARCHITECTURE 52 7.8 STEP 1.7 DETERMINE ENTERPRISE ROADMAP 55 7.8 STEP 1.7 DETERMINE ENTERPRISE ROADMAP			
5.2 ARCHITECTURE MODELS 26 6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE VALUE STREAM 31 6.2 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE 35 6.4 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE 41 7.1 THE STEPS 41 7.2 STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY. 41 7.3 STEP 1.2 DETERMINE BUSINESS ARCHITECTURE 48 7.5 STEP 1.3 DETERMINE BUSINESS ARCHITECTURE 48 7.5 STEP 1.4 DETERMINE INFORMATION ARCHITECTURE 50 7.6 STEP 1.5 DETERMINE INFRASTRUCTURE ARCHITECTURE 52 7.7 STEP 1.6 DETERMINE INFRASTRUCTURE ARCHITECTURE 52 7.8 STEP 1.7 DETERMINE BAR AND CHANGE STRATEGY 55 7.8 STEP 1.1 DETERMINE HIFERPRISE ROADMAP 56 7.9 STEP 2.1 DETERMINE HIGH LEVEL REQUIREMENTS 56 7.10 STEP 2.1 DETERMINE HIGH LEVEL			
6 CONTINUOUS ARCHITECTURE DESIGN 31 6.1 CONTINUOUS ARCHITECTURE VALUE STREAM 31 6.2 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR 31 6.3 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE 35 6.4 CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS 37 7 CONTINUOUS ARCHITECTURE 41 7.1 THE STEPS 41 7.2 STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY. 41 7.3 STEP 1.2 DETERMINE BUSINESS ARCHITECTURE 44 7.4 STEP 1.3 DETERMINE BUSINESS ARCHITECTURE 48 7.5 STEP 1.4 DETERMINE INFORMATION ARCHITECTURE 50 7.6 STEP 1.5 DETERMINE APPLICATION ARCHITECTURE 52 7.7 STEP 1.6 DETERMINE APPLICATION ARCHITECTURE 52 7.8 STEP 1.7 DETERMINE ENTERPRISE ROADMAP 56 7.9 STEP 2.1 DETERMINE HIGH LEVEL REQUIREMENTS 56 7.10 STEP 2.2 DETERMINE PRODUCT VISION 56 7.11 STEP 2.3 PRODUCT ROADMAP 57 7.12 STEP 2.4 PROVIDE DIRECTION AND MONITOR IT 58 7.13 STEP 3.1 DETERMINE VALUE STREAMS			
6.1CONTINUOUS ARCHITECTURE VALUE STREAM316.2CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR316.3CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE356.4CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS377CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS377CONTINUOUS ARCHITECTURE417.1THE STEPS417.2STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY417.3STEP 1.2 DETERMINE BUSINESS ARCHITECTURE447.4STEP 1.3 DETERMINE INFORMATION ARCHITECTURE487.5STEP 1.4 DETERMINE APPLICATION ARCHITECTURE507.6STEP 1.5 DETERMINE INFRASTRUCTURE ARCHITECTURE527.7STEP 1.6 DETERMINE GAPS AND CHANGE STRATEGY557.8STEP 1.7 DETERMINE ENTERPRISE ROADMAP567.9STEP 2.1 DETERMINE HIGH LEVEL REQUIREMENTS567.10STEP 2.2 DETERMINE HIGH LEVEL REQUIREMENTS567.11STEP 2.3 PRODUCT ROADMAP577.12STEP 2.4 PROVIDE DIRECTION AND MONITOR IT587.13STEP 3.1 DETERMINE VALUE SYSTEMS587.14STEP 3.2 DETERMINE VALUE STREAMS587.15STEP 3.3 DETERMINE SERVICES58	5.2	Architecture Models	26
6.1CONTINUOUS ARCHITECTURE VALUE STREAM316.2CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR316.3CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE356.4CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS377CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS377CONTINUOUS ARCHITECTURE417.1THE STEPS417.2STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY417.3STEP 1.2 DETERMINE BUSINESS ARCHITECTURE447.4STEP 1.3 DETERMINE INFORMATION ARCHITECTURE487.5STEP 1.4 DETERMINE APPLICATION ARCHITECTURE507.6STEP 1.5 DETERMINE INFRASTRUCTURE ARCHITECTURE527.7STEP 1.6 DETERMINE GAPS AND CHANGE STRATEGY557.8STEP 1.7 DETERMINE ENTERPRISE ROADMAP567.9STEP 2.1 DETERMINE HIGH LEVEL REQUIREMENTS567.10STEP 2.2 DETERMINE HIGH LEVEL REQUIREMENTS567.11STEP 2.3 PRODUCT ROADMAP577.12STEP 2.4 PROVIDE DIRECTION AND MONITOR IT587.13STEP 3.1 DETERMINE VALUE SYSTEMS587.14STEP 3.2 DETERMINE VALUE STREAMS587.15STEP 3.3 DETERMINE SERVICES58	6	CONTINUOUS ARCHITECTURE DESIGN	31
6.2CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOR316.3CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE356.4CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS377CONTINUOUS ARCHITECTURE417.1THE STEPS417.2STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY417.3STEP 1.2 DETERMINE BUSINESS ARCHITECTURE447.4STEP 1.3 DETERMINE BUSINESS ARCHITECTURE447.5STEP 1.3 DETERMINE INFORMATION ARCHITECTURE507.6STEP 1.4 DETERMINE INFORMATION ARCHITECTURE507.6STEP 1.5 DETERMINE INFRASTRUCTURE ARCHITECTURE527.7STEP 1.6 DETERMINE GAPS AND CHANGE STRATEGY557.8STEP 1.7 DETERMINE ENTERPRISE ROADMAP567.9STEP 2.1 DETERMINE HIGH LEVEL REQUIREMENTS567.10STEP 2.2 DETERMINE PRODUCT VISION567.11STEP 2.3 PRODUCT ROADMAP577.12STEP 3.1 DETERMINE VALUE SYSTEMS587.14STEP 3.2 DETERMINE VALUE STREAMS587.15STEP 3.3 DETERMINE SERVICES58	6.1	Continuous Architecture Value Stream	31
6.4CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS.377CONTINUOUS ARCHITECTURE.417.1THE STEPS.417.2STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY.417.3STEP 1.2 DETERMINE BUSINESS ARCHITECTURE447.4STEP 1.3 DETERMINE INFORMATION ARCHITECTURE487.5STEP 1.4 DETERMINE APPLICATION ARCHITECTURE507.6STEP 1.5 DETERMINE INFRASTRUCTURE ARCHITECTURE527.7STEP 1.6 DETERMINE GAPS AND CHANGE STRATEGY557.8STEP 1.7 DETERMINE ENTERPRISE ROADMAP567.9STEP 2.1 DETERMINE HIGH LEVEL REQUIREMENTS.567.10STEP 2.2 DETERMINE PRODUCT VISION567.11STEP 2.3 PRODUCT ROADMAP577.12STEP 2.4 PROVIDE DIRECTION AND MONITOR IT587.13STEP 3.1 DETERMINE VALUE STREAMS587.14STEP 3.2 DETERMINE VALUE STREAMS587.15STEP 3.3 DETERMINE SERVICES58	6.2		
7CONTINUOUS ARCHITECTURE417.1THE STEPS417.2STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY.417.3STEP 1.2 DETERMINE BUSINESS ARCHITECTURE447.4STEP 1.3 DETERMINE INFORMATION ARCHITECTURE487.5STEP 1.4 DETERMINE APPLICATION ARCHITECTURE507.6STEP 1.5 DETERMINE INFRASTRUCTURE ARCHITECTURE507.6STEP 1.5 DETERMINE GAPS AND CHANGE STRATEGY557.8STEP 1.7 DETERMINE ENTERPRISE ROADMAP567.9STEP 2.1 DETERMINE HIGH LEVEL REQUIREMENTS.567.10STEP 2.2 DETERMINE PRODUCT VISION.567.11STEP 2.3 PRODUCT ROADMAP577.12STEP 3.1 DETERMINE VALUE SYSTEMS587.13STEP 3.1 DETERMINE VALUE SYSTEMS587.14STEP 3.2 DETERMINE VALUE STREAMS587.15STEP 3.3 DETERMINE SERVICES58	6.3	CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOE	35
7.1THE STEPS.417.2STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY.417.3STEP 1.2 DETERMINE BUSINESS ARCHITECTURE447.4STEP 1.3 DETERMINE INFORMATION ARCHITECTURE487.5STEP 1.4 DETERMINE APPLICATION ARCHITECTURE507.6STEP 1.5 DETERMINE INFRASTRUCTURE ARCHITECTURE527.7STEP 1.6 DETERMINE GAPS AND CHANGE STRATEGY557.8STEP 1.7 DETERMINE ENTERPRISE ROADMAP567.9STEP 2.1 DETERMINE HIGH LEVEL REQUIREMENTS.567.10STEP 2.2 DETERMINE PRODUCT VISION567.11STEP 2.3 PRODUCT ROADMAP577.12STEP 3.1 DETERMINE VALUE SYSTEMS587.13STEP 3.1 DETERMINE VALUE STREAMS.587.14STEP 3.3 DETERMINE SERVICES58	6.4	CONTINUOUS ARCHITECTURE USE CASE DIAGRAM SOS	37
7.1THE STEPS.417.2STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY.417.3STEP 1.2 DETERMINE BUSINESS ARCHITECTURE447.4STEP 1.3 DETERMINE INFORMATION ARCHITECTURE487.5STEP 1.4 DETERMINE APPLICATION ARCHITECTURE507.6STEP 1.5 DETERMINE INFRASTRUCTURE ARCHITECTURE527.7STEP 1.6 DETERMINE GAPS AND CHANGE STRATEGY557.8STEP 1.7 DETERMINE ENTERPRISE ROADMAP567.9STEP 2.1 DETERMINE HIGH LEVEL REQUIREMENTS.567.10STEP 2.2 DETERMINE PRODUCT VISION567.11STEP 2.3 PRODUCT ROADMAP577.12STEP 3.1 DETERMINE VALUE SYSTEMS587.13STEP 3.1 DETERMINE VALUE STREAMS.587.14STEP 3.3 DETERMINE SERVICES58	7		11
7.2STEP 1.1 DETERMINE BUSINESS GOALS AND STRATEGY.417.3STEP 1.2 DETERMINE BUSINESS ARCHITECTURE447.4STEP 1.3 DETERMINE INFORMATION ARCHITECTURE.487.5STEP 1.4 DETERMINE APPLICATION ARCHITECTURE507.6STEP 1.5 DETERMINE INFRASTRUCTURE ARCHITECTURE527.7STEP 1.6 DETERMINE GAPS AND CHANGE STRATEGY557.8STEP 1.7 DETERMINE ENTERPRISE ROADMAP567.9STEP 2.1 DETERMINE HIGH LEVEL REQUIREMENTS.567.10STEP 2.2 DETERMINE PRODUCT VISION567.11STEP 2.3 PRODUCT ROADMAP577.12STEP 2.4 PROVIDE DIRECTION AND MONITOR IT587.13STEP 3.1 DETERMINE VALUE SYSTEMS587.14STEP 3.2 DETERMINE VALUE STREAMS587.15STEP 3.3 DETERMINE SERVICES58	-		
7.3STEP 1.2 DETERMINE BUSINESS ARCHITECTURE447.4STEP 1.3 DETERMINE INFORMATION ARCHITECTURE487.5STEP 1.4 DETERMINE APPLICATION ARCHITECTURE507.6STEP 1.5 DETERMINE INFRASTRUCTURE ARCHITECTURE527.7STEP 1.6 DETERMINE GAPS AND CHANGE STRATEGY557.8STEP 1.7 DETERMINE ENTERPRISE ROADMAP567.9STEP 2.1 DETERMINE HIGH LEVEL REQUIREMENTS567.10STEP 2.2 DETERMINE PRODUCT VISION567.11STEP 2.3 PRODUCT ROADMAP577.12STEP 2.4 PROVIDE DIRECTION AND MONITOR IT587.13STEP 3.1 DETERMINE VALUE SYSTEMS587.14STEP 3.2 DETERMINE VALUE STREAMS587.15STEP 3.3 DETERMINE SERVICES58			
7.4STEP 1.3 DETERMINE INFORMATION ARCHITECTURE.487.5STEP 1.4 DETERMINE APPLICATION ARCHITECTURE.507.6STEP 1.5 DETERMINE INFRASTRUCTURE ARCHITECTURE527.7STEP 1.6 DETERMINE GAPS AND CHANGE STRATEGY557.8STEP 1.7 DETERMINE ENTERPRISE ROADMAP567.9STEP 2.1 DETERMINE HIGH LEVEL REQUIREMENTS.567.10STEP 2.2 DETERMINE PRODUCT VISION567.11STEP 2.3 PRODUCT ROADMAP577.12STEP 2.4 PROVIDE DIRECTION AND MONITOR IT587.13STEP 3.1 DETERMINE VALUE SYSTEMS587.14STEP 3.2 DETERMINE VALUE STREAMS587.15STEP 3.3 DETERMINE SERVICES58			
7.5STEP 1.4 DETERMINE APPLICATION ARCHITECTURE.507.6STEP 1.5 DETERMINE INFRASTRUCTURE ARCHITECTURE527.7STEP 1.6 DETERMINE GAPS AND CHANGE STRATEGY557.8STEP 1.7 DETERMINE ENTERPRISE ROADMAP567.9STEP 2.1 DETERMINE HIGH LEVEL REQUIREMENTS.567.10STEP 2.2 DETERMINE PRODUCT VISION567.11STEP 2.3 PRODUCT ROADMAP577.12STEP 2.4 PROVIDE DIRECTION AND MONITOR IT587.13STEP 3.1 DETERMINE VALUE SYSTEMS587.14STEP 3.2 DETERMINE VALUE STREAMS587.15STEP 3.3 DETERMINE SERVICES58			
7.6Step 1.5 Determine infrastructure architecture527.7Step 1.6 Determine GAPS and change strategy557.8Step 1.7 Determine enterprise roadmap567.9Step 2.1 Determine high level requirements567.10Step 2.2 Determine product vision567.11Step 2.3 Product roadmap577.12Step 2.4 Provide direction and monitor it587.13Step 3.1 Determine value systems587.14Step 3.2 Determine value streams587.15Step 3.3 Determine services58			
7.7STEP 1.6 DETERMINE GAPS AND CHANGE STRATEGY557.8STEP 1.7 DETERMINE ENTERPRISE ROADMAP567.9STEP 2.1 DETERMINE HIGH LEVEL REQUIREMENTS567.10STEP 2.2 DETERMINE PRODUCT VISION567.11STEP 2.3 PRODUCT ROADMAP577.12STEP 2.4 PROVIDE DIRECTION AND MONITOR IT587.13STEP 3.1 DETERMINE VALUE SYSTEMS587.14STEP 3.2 DETERMINE VALUE STREAMS587.15STEP 3.3 DETERMINE SERVICES58			
7.8Step 1.7 Determine enterprise roadmap567.9Step 2.1 Determine high level requirements567.10Step 2.2 Determine product vision567.11Step 2.3 Product roadmap577.12Step 2.4 Provide direction and monitor it587.13Step 3.1 Determine value systems587.14Step 3.2 Determine value streams587.15Step 3.3 Determine services58			
7.9Step 2.1 Determine high level requirements.567.10Step 2.2 Determine product vision.567.11Step 2.3 Product roadmap577.12Step 2.4 Provide direction and monitor it587.13Step 3.1 Determine value systems587.14Step 3.2 Determine value streams.587.15Step 3.3 Determine services58			
7.10STEP 2.2 DETERMINE PRODUCT VISION567.11STEP 2.3 PRODUCT ROADMAP577.12STEP 2.4 PROVIDE DIRECTION AND MONITOR IT587.13STEP 3.1 DETERMINE VALUE SYSTEMS587.14STEP 3.2 DETERMINE VALUE STREAMS587.15STEP 3.3 DETERMINE SERVICES58			
7.11STEP 2.3 PRODUCT ROADMAP577.12STEP 2.4 PROVIDE DIRECTION AND MONITOR IT587.13STEP 3.1 DETERMINE VALUE SYSTEMS587.14STEP 3.2 DETERMINE VALUE STREAMS587.15STEP 3.3 DETERMINE SERVICES58			
7.12Step 2.4 Provide direction and monitor it587.13Step 3.1 Determine value systems587.14Step 3.2 Determine value streams587.15Step 3.3 Determine services58			
7.13 Step 3.1 Determine value systems			
7.14 Step 3.2 Determine value streams			
7.15 STEP 3.3 DETERMINE SERVICES			
	7.16		
	7.13	STEP 3.1 DETERMINE VALUE SYSTEMS	58

VIII DevOps Continuous Architecture

7.17		
7.18	STEP 3.6 DETERMINE SERVICE ROADMAP	
8 0	CONTINUOUS PLANNING	61
8.1	CONTINUOUS PLANNING DEFINITION	
8.2	CONTINUOUS PLANNING VALUE STREAM	
8.3	RELATION WITH CONTINUOUS ARCHITECTURE	
8.4	ARCHITECTURE PRINCIPLES	
8.5	ARCHITECTURE MODELS	
9 (CONTINUOUS SLA	
9.1	CONTINUOUS SLA DEFINITION	
9.2	CONTINUOUS SLA VALUE STREAM	
9.3	RELATION WITH CONTINUOUS ARCHITECTURE	
9.4	Architecture Principles	
9.5	ARCHITECTURE MODELS	
10 0	CONTINUOUS DESIGN	77
10.1	CONTINUOUS DESIGN DEFINITION	
10.2		
10.3		
10.4	Architecture Principles	
10.5	Architecture Models	
11 0	CONTINUOUS ACCEPTANCE	83
11.1	CONTINUOUS ACCEPTANCE DEFINITION	
11.2	CONTINUOUS ACCEPTANCE VALUE STREAM	
11.3	Relation with Continuous Architecture	
11.4	ARCHITECTURE PRINCIPLES	
11.5	Architecture Models	
12 0	CONTINUOUS TESTING	
12.1	Continuous Testing definition	
12.2	CONTINUOUS TESTING VALUE STREAM	
12.3	Relation with Continuous Architecture	
12.4	Architecture Principles	
12.5	Architecture Models	
13 0	CONTINUOUS INTEGRATION	
13.1	CONTINUOUS INTEGRATION DEFINITION	
13.2	CONTINUOUS INTEGRATION VALUE STREAM	
13.3	RELATION WITH CONTINUOUS ARCHITECTURE	
13.4		
13.5	Architecture Models	
14 C		
14.1	CONTINUOUS AI-DEFINITION	
14.2		
14.3		
14.4		
14.5	ARCHITECTURE MODELS	
15 C	CONTINUOUS DEPLOYMENT	
15.1	CONTINUOUS DEPLOYMENT DEFINITION	

15.2	CONTINUOUS DEPLOYMENT VALUE STREAM	125
15.3	RELATION WITH CONTINUOUS ARCHITECTURE	
15.4	ARCHITECTURE PRINCIPLES	
15.5	Architecture Models	129
16 CC	ONTINUOUS MONITORING	133
16.1	Continuous Monitoring definition	
16.2	Continuous Monitoring value stream	
16.3	RELATION WITH CONTINUOUS ARCHITECTURE.	
16.4	Architecture Principles	134
16.5	Architecture Models	138
17 CC	ONTINUOUS LEARNING	145
17.1	Continuous Learning definition	
17.1	CONTINUOUS LEARNING DEFINITION	
17.3	RELATION WITH CONTINUOUS ARCHITECTURE	
17.4	Architecture Principles	
17.5	Architecture Models	
18 CC	ONTINUOUS ASSESSMENT	155
18.1	Continuous Assessment definition	
18.1	CONTINUOUS ASSESSMENT DEFINITION	
18.2	RELATION WITH CONTINUOUS ARCHITECTURE.	
18.3	ARCHITECTURE PRINCIPLES	
18.5	Architecture Models	
	ONTINUOUS AUDITING	
19 00	Continuous Auditing definition.	
19.1	CONTINUOUS AUDITING VALUE STREAM	
19.2	RELATION WITH CONTINUOUS ARCHITECTURE.	
19.4	Architecture Principles	
19.5	ARCHITECTURE MODELS.	
20 CC	ONTINUOUS SECURITY	175
20.1	CONTINUOUS SECURITY DEFINITION	
20.2	CONTINUOUS SECURITY VALUE STREAM	
20.3	RELATION WITH CONTINUOUS ARCHITECTURE	
20.4	Architecture Principles	
20.5	Architecture Models.	
21 CC	ONTINUOUS ARCHITECTURE ASSESSMENT	180
21.1	21.1 WHAT IS THE CE MODEL?	
21.2	MATURITY DIMENSIONS	
21.3	DEVOPS CE-MODEL, CH	
	DIX A, LITERATURE LIST	
APPEN	DIX B, GLOSSARY	201
APPEN	DIX C, ABBREVIATIONS	217
APPEN	DIX D, WEBSITES	223
APPEN	DIX E, INDEX	225

Figures

_	
FIGURE 1-1, DEVOPS LEMNISCATE	1
FIGURE 2-1, SOR AND SOE, SOURCE HSO THE RESULT COMPANY.	5
FIGURE 2-2, CONTINUOUS ARCHITECTURE MODEL SOR.	6
FIGURE 2-3, ROADMAP TO VALUE MODEL [LAYTON 2017].	6
FIGURE 2-4, ENTERPRISE ARCHITECTURE	7
FIGURE 2-5, ROADMAP	8
FIGURE 2-6, VALUE CHAIN OF PORTER, SOURCE: [BOOK MICHAEL PORTER].	9
FIGURE 4-1, CHANGE PARADIGM.	15
FIGURE 4-2, CHANGE PARADIGM - VISION.	16
FIGURE 4-3, CHANGE PARADIGM - POWER.	
FIGURE 4-4, CHANGE PARADIGM - ORGANISATION	20
FIGURE 4-5, CHANGE PARADIGM - RESOURCES	
FIGURE 5-1, CONTINUOUS ARCHITECTURE MODEL	
FIGURE 5-2, ROADMAP TO VALUE, SOURCE: [LAYTON 2017].	27
FIGURE 5-3, TEMPLATE SYSTEM BUILDING BLOCKS – COMMON.	28
FIGURE 6-1, CONTINUOUS ARCHITECTURE VALUE STREAM.	31
FIGURE 6-2, USE CASE DIAGRAM FOR UPFRONT DESIGN SOR.	32
FIGURE 6-3, USE CASE DIAGRAM FOR UPFRONT DESIGN SOE.	35
FIGURE 6-4, USE CASE DIAGRAM FOR SYSTEM OF SERVICES.	38
FIGURE 7-1, CONTINUOUS ARCHITECTURE VALUE STREAM.	41
FIGURE 7-2, SWOT.	42
FIGURE 7-3, BALANCED SCORECARD [KAPLAN 2004]	42
FIGURE 7-4, CASCADED BALANCED SCORECARD.	43
FIGURE 7-5, BUSINESS MODEL CANVAS.	44
FIGURE 7-6, VALUE CHAIN OF PORTER, SOURCE: [BOOK MICHAEL PORTER].	45
FIGURE 7-7, RECURSIEVE VALUE CHAIN OF PORTER, SOURCE: [BOOK MICHAEL PORTER]	45
FIGURE 7-8, RECURSIEVE VALUE CHAIN OF PORTER, SOURCE: [BOOK MICHAEL PORTER]	
FIGURE 7-9, SERVICE PORTFOLIO.	47
FIGURE 7-10, ARCHIMATE WITH VALUE STREAMS.	47
FIGURE 7-11, ARCHIMATE WITH INFORMATION SERVICES.	48
FIGURE 7-12, TEMPLATE SYSTEM BUILDING BLOCKS – INFORMATION.	
FIGURE 7-13, ARCHIMATE WITH APPLICATION SERVICES.	
FIGURE 7-14, TEMPLATE SYSTEM BUILDING BLOCKS – APPLICATION.	
FIGURE 7-15, ARCHIMATE INFRASTRUCTURE SERVICES.	
FIGURE 7-16, TEMPLATE SYSTEM BUILDING BLOCKS – TECHNOLOGY.	
FIGURE 7-17, VALUE STREAM MAPPING INFRASTRUCTURE.	54
FIGURE 7-18, VALUE STREAM CANVAS TEMPLATE.	
FIGURE 8-1, CONTINUOUS PLANNING	
FIGURE 8-2, ROADMAP TO VALUE, BRON: [LAYTON 2017].	
FIGURE 8-3, CONTINUOUS PLANNING MODEL.	
FIGURE 8-4, PLANNING & DESIGN MODEL	
FIGURE 9-1, CONTINUOUS SLA.	
FIGURE 9-2, ROADMAP TO VALUE, SOURCE: [LAYTON 2017].	
FIGURE 9-3, CONTINUOUS SLA MODEL.	
FIGURE 9-4, VALUE STREAM MAPPING MODEL.	
FIGURE 10-1, CONTINUOUS DESIGN.	
FIGURE 10-2, CONTINUOUS DESIGN PYRAMID WITH DELIVERABLES AND QUESTIONS TO BE ANSWERED.	
FIGURE 10-3, CONTINUOUS DESIGN PLANNING MODEL.	
FIGURE 11-1, CONTINUOUS ACCEPTANCE	
FIGURE 11-2, VALUE STREAM CANVAS TEMPLATE.	

FIGURE 11-3, TEMPLATE SYSTEM BUILDING BLOCKS – COMMON.	89
FIGURE 11-4, TEMPLATE SYSTEM BUILDING BLOCKS – INFORMATION.	90
FIGURE 11-5, TEMPLATE SYSTEM BUILDING BLOCKS – APPLICATION.	91
FIGURE 11-6, TEMPLATE SYSTEM BUILDING BLOCKS – TECHNOLOGY.	93
FIGURE 12-1, CONTINUOUS TESTING.	95
FIGURE 12-2, IDEAL TEST PYRAMID.	99
FIGURE 12-3, NON-IDEAL TEST PYRAMID.	100
FIGURE 13-1, CONTINUOUS INTEGRATION.	105
FIGURE 13-2, VERSION CONTROL CLASS MODEL.	109
FIGURE 13-3, LOCAL BACKUP VERSION CONTROL.	
FIGURE 13-4, LOCAL MANUAL VERSION CONTROL.	110
FIGURE 13-5, LOCAL VERSION CONTROL SYSTEM.	111
FIGURE 13-6, CENTRAL VERSION MANAGEMENT SYSTEM.	
FIGURE 13-7, DISTRIBUTED VERSION CONTROL SYSTEM.	
FIGURE 14-1, CONTINUOUS AI.	117
FIGURE 14-2, AI PORTFOLIO.	122
FIGURE 14-3, AI-PATTERN LIBRARY.	
FIGURE 15-1, CONTINUOUS DEPLOYMENT.	
FIGURE 15-2, CONTINUOUS DEPLOYMENT ROADMAP.	
FIGURE 16-1, CONTINUOUS MONITORING.	
FIGURE 16-2, CONTINUOUS MONITORING GOVERNANCE MODEL.	
FIGURE 16-3, CONTINUOUS MONITORING LAYER MODEL.	
FIGURE 16-4, LEAD AND LAG PERFORMANCE INDICATORS MANAGEMENT MODEL.	
FIGURE 16-5, MONITOR HIERARCHY MODEL.	
FIGURE 17-1, CONTINUOUS LEARNING.	
FIGURE 17-2, CONTINUOUS LEARNING MODEL.	
FIGURE 17-3, I-T-E SHAPED MODEL.	
FIGURE 17-4, HIGH PERFORMANCE MODEL BASED ON [WESTRUM].	
FIGURE 18-1, CONTINUOUS ASSESSMENT.	
FIGURE 18-2, NECKER CUBE.	
FIGURE 18-3, FRONT DEVOPS CUBE.	
FIGURE 18-4, REAR DEVOPS CUBE	
FIGURE 18-5, DEVOPS CE-SPIDER MODEL	
FIGURE 19-1, CONTINUOUS AUDITING.	
Figure 19-2, Continuous Auditing Pyramid.	
FIGURE 19-3, CONTINUOUS AUDITING PYRAMID DEPICTED ON THE DEVOPS LEMNISCATE.	
FIGURE 19-4, CONTINUOUS AUDITING PYRAMID WITH DELIVERABLES AND QUESTIONS TO BE ANSWERED	
FIGURE 19-5, CONTINUOUS AUDITING PYRAMID MODEL MAPPED ONTO CONTINUOUS CONTROL MODEL	
FIGURE 19-6, QUALITY CONTROL & ASSURANCE MODEL	
FIGURE 20-1, CONTINUOUS SECURITY.	
FIGURE 20-2, CONTINUOUS SECURITY PYRAMID.	
FIGURE 20-3, CONTINUOUS SECURITY PYRAMID DEPICTED ON THE DEVOPS LEMNISCATE.	
FIGURE 20-4, CONTINUOUS SECURITY PYRAMID WITH DELIVERABLES AND QUESTIONS TO BE ANSWERED	
FIGURE 20-5, CONTINUOUS SECURITY PYRAMID MODEL MAPPED ONTO CONTINUOUS CONTROL MODEL	
FIGURE 20-6, QUALITY CONTROL & ASSURANCE MODEL	
FIGURE 20-7, RECURSIVE VALUE CHAIN.	
FIGURE 20-8, INFORMATION SECURITY VALUE CHAIN.	
FIGURE 20-9, INFORMATION SECURITY VALUE SYSTEM.	
FIGURE 20-10, INFORMATION SECURITY PRACTICES.	
FIGURE 20-11, INFORMATION SECURITY VALUE SYSTEM OVERVIEW.	
FIGURE 20-12, SERVICE VALUE CHAIN.	

FIGURE 20-13, DEVELOPMENT VALUE CHAIN.	186
FIGURE 20-14, CONTINUOUS SECURITY PYRAMID DEPICTED ON THE ISVS, DVS AND SVS MODELS	187
FIGURE 20-15, INFORMATION SECURITY PERSPECTIVES.	188
FIGURE 21-1, DEVOPS CE-SPIDER MODEL.	191
FIGURE 21-2, DEVOPS CH-SPIDER MODEL	194

Tables

TABLE 1-1, CONTINUOUS EVERYTHING ASPECTS.	2
TABLE 1-2, APPENDICES.	3
TABLE 2-1, PLANNING OBJECTS.	8
TABLE 3-1, COMMON PROBLEMS WHEN USING CONTINUOUS ARCHITECTURE.	
TABLE 6-1, USE CASE TEMPLATE.	
TABLE 6-2, Use Case for Continuous Architecture SOR.	
TABLE 6-3, Use Case for Continuous Architecture SOS	
TABLE 6-4, Use Case for Continuous Architecture SOS	
TABLE 7-1, TEMPLATE OF A ROADMAP.	57
TABLE 7-2, TEMPLATE OF AN EPIC ONE PAGER	58
TABLE 12-1, TEST TYPE MATRIX TEMPLATE.	
TABLE 12-2, TEST TYPE MATRIX EXAMPLE.	
TABLE 12-3, TEST TECHNOLOGY MATRIX TEMPLATE.	
TABLE 12-4, TEST TECHNIQUE-MATRIX EXAMPLE.	
TABLE 12-5, TEST OBJECT-MATRIX.	
TABLE 12-6, TEST OBJECT-MATRIX EXAMPLE.	
TABLE 12-7, TEST TOOL-MATRIX PATTERN.	
TABLE 12-8, TEST TOOL-MATRIX EXAMPLE.	
TABLE 13-1, FEATURES OF VERSION CONTROL SYSTEMS.	
TABLE 13-2, OTHER FEATURES OF VERSION CONTROL SYSTEMS.	
TABLE 13-3, RISKS OF VERSION CONTROL SYSTEMS.	
TABLE 13-4, AN OVERVIEW OF SOME VERSION CONTROL SYSTEMS.	116
TABLE 16-1, EVENT CORRELATION WITH AI.	
TABLE 17-1, CHARACTERISTICS AND BEHAVIORS OF I-T-E SHAPED PEOPLE.	
TABLE 18-1, CE MATURITY MODEL.	
TABLE 18-2, CONTINUOUS EVERYTHING.	
TABLE 18-3, CMMI LEVELS FOR CONTINUOUS EVERYTHING.	
TABLE 18-4, PRINCIPLE OF MATURITY LEVELS.	
TABLE 21-1, DEVOPS CE-MODEL.	
TABLE 21-2, CONTINUOUS EVERYTHING.	
TABLE 21-3, CMMI LEVELS FOR CONTINUOUS EVERYTHING.	
TABLE 21-4, PR-ORG-009. MATURITY LEVELS.	
TABLE 21-5, CH MATURITY CHARACTERISTICS.	

Appendices

Appendix A, Literature list	.197
APPENDIX B, GLOSSARY	.201
APPENDIX C, ABBREVIATIONS	.217
APPENDIX D, WEBSITES	
Appendix E, Index	
,	

Preface

This book has been compiled based on my experiences with Continuous Architecture. The DevOps (CE) aspect area Continuous Architecture includes providing direction for the development and operation of the information systems and DevOps value streams. This direction is based on the mission, vision and strategy of the organisation and is translated by architects into an IST, SOLL and migration path based on architectural principles and models.

This direction provides top-down control for the Agile programs and projects. This aspect area of Continuous Everything is closely related to Continuous Planning and Continuous Design. The main message in this book is to tailor the Continuous Architecture value stream to three application areas. Where the information provision to be realised is part of a larger whole, such as a chain of information systems (System of Records), a preliminary architectural analysis is required (upfront), followed by a growth architecture (emerging). Where information systems are decoupled from other information systems and are therefore part of a chain of information systems (System of Engagement), an emerging architecture can be directly envisaged. Continuous Architecture thus supports the architectural implementation of the Business Value System (BVS). A value system is defined in this book as the management of a value chain that creates value.

But not only the business value streams need to be supported with an information provision, the DevOps value streams also require the design of an information provision. In this book this is called the System of Services. This SOS refers to the architectural development and management of the Service Value System (ITIL 4), the Development Value System (Agile Scrum, Kanban et cetera) and the Information Security Value System (ISO 27001). With regard to the management of business value streams, the BVS also falls under this.

The role of architecture in DevOps teams has changed in many organisations compared to the traditional working method. The collaboration between the architects and the DevOps engineers is much more intensive, more frequent and the architects become more of a coach.

Partly in view of the speed at which the world of DevOps is developing and the need to give you as many images as possible with as little text as possible about how to deal with Continuous Architecture in the world of DevOps, I have decided to keep this book Agile. This means that it briefly describes what are important options for applying architectural principles and models.

I have also shared many of my experiences in the articles on <u>www.ITpedia.nl</u>. I have also translated the knowledge and skills into various training courses that I provide. These can be found at <u>www.dbmetrics.nl</u>.

I would like to thank the following people for their inspiring contribution to this book and the wonderful collaboration!

٠	D. (Dennis) Boersen	Argis IT Consultants
٠	F. (Freek) de Cloe	smartdocs.com
٠	J.A.E. (Jane) ten Have	-
•	Dr. L.J.G.T. (Louis) van Hemmen	BitAll B.V.
٠	J.W. (Jan-Willem) Hordijk	Digital Transformation Advisor, Dutch Nordic Sweden
٠	W. (Willem) Kok	Argis IT Consultants
:	M.I. (Mohammed Ismail) Mustapha	NV Haagse Milieu Services
•	F (Franklin) Selgert	-
٠	N (Niels) Talens	www.nielstalens.nl

I hope you enjoy reading this book and, above all, good luck in applying Continuous Architecture within your own organisation.

If you have any questions or comments, please do not hesitate to contact me. A lot of time has been spent making this book as complete and consistent as possible.

XVI DevOps Continuous Architecture

If you find any shortcomings, I would appreciate it if you would inform me so that these matters can be processed in the next edition.

1 Introduction

Reading Guide:

This chapter describes the purpose of this book (1.1), the intended target group (1.2), the background of this book (1.3), the structure (1.4), the appendices (1.5) and finally some tips for using this book in section 1.6.

1.1 Objective

The objective of this book is to provide basic knowledge of Continuous Architecture and tips and tricks for applying this aspect area of Continuous Everything.

1.2 Target group

The target group of this book are all functions involved in the DevOps teams. This includes architects, marketers, buyers, line managers, Dev engineers, Ops engineers, Product owners, Scrum masters, Agile Coaches, and representatives of the user organisation. This book is of course also very suitable for process owners, process managers, et cetera who are involved in the development of information provision using a DevOps method. Finally, there is a target group that does not develop or manage, but which determines whether the value streams meet the required criteria such as legislation and regulations. This target group includes quality control, staff, and auditors. They can use this book to identify gaps in risk management that need to be accepted or managed.

1.3 Background

This book contains various considerations to continuously shape the knowledge and skills development of DevOps engineers. Continuous Architecture is part of the DevOps Lemniscate as shown in Figure 1-1.

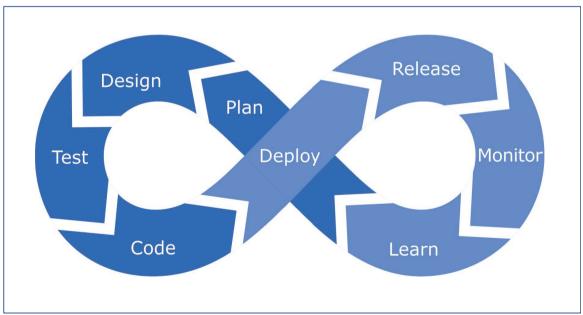


Figure 1-1, DevOps Lemniscate.

The DevOps Lemniscate provides an overview of the phases to be completed to continuously produce software. The DevOps Lemniscate is therefore a good basis for defining the concept of Continuous Architecture.

Continuous Architecture is one of the aspect areas of the Continuous Everything concept. The CE concept describes all phases of the DevOps Lemniscate in the form of continuous activities to be performed. Table 1-1 shows the relationship between the steps of the DevOps Lemniscate and the Continuous Everything aspect areas.

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

Table 1-1, Continuous Everything aspects.

Continuous Auditing (9) and Continuous Assessment (10) are not shown in the DevOps Lemniscate, nor are other Continuous aspect areas such as Continuous Architecture for the sake of simplicity of the DevOps Lemniscate.

The word 'Continuous' refers to the continuous gathering of knowledge, skills, and data in order to distil and apply improvements and applications. First, there must be continuous learning during the execution of Agile projects. This can be done by building a control loop into the development process, such as the 'inspect and adapt' moments in the Agile Scrum approach and the Lean principle of validated learning. But in addition, the knowledge and skill levels of the DevOps engineers must also be continuously aligned with the to be achieved goals in the value streams. For Continuous Architecture, the concept of Continuous mainly refers to the cyclical analysis of the business mission, vision and strategy to determine the current situation (IST), future situation (SOLL) and the migration path to get from IST to SOLL in order to realise the strategy, including the information provision of the service organisation in the sense of Continuous Everything value streams in the various value systems.

To successfully achieve the business strategy, the architects must provide direction for the design and operation of the Continuous Everything aspect areas in the form of architectural principles and architectural models.

1.4 Structure

This book describes how to shape Continuous Architecture top-down from an organisation's strategy. Before this approach is discussed, the definitions, anchoring and architecture of Continuous Architecture are first discussed. This is followed by the steps that need to be completed in a number of chapters.

1.4.1 Chapter 2: Basic concepts en basic terms

This chapter discusses the basic concepts and basic terms of Continuous Architecture. General concepts are also described such as the value chain, value streams, value systems and types of information systems.

1.4.2 Chapter 3: Continuous Architecture definition

It is important to have a general definition of Continuous Architecture. Therefore, this chapter provides a definition of this concept and discusses the problems and possible causes of Continuous Architecture not getting off the ground.

1.4.3 Chapter 4: Continuous Architecture anchoring

This chapter discusses how Continuous Architecture can be anchored through the change paradigm. The following questions are answered:

- What is the vision on Continuous Architecture (Vision)?
- Where are the responsibilities and authorities (Power)?
- How can Continuous Architecture be applied (Organisation)?
- Which profiles of people and which resources are needed (Resources)?

1.4.4 Chapter 5: Continuous Architecture architecture

This chapter describes the architectural principles and models for Continuous Architecture. The architecture models include the Continuous Architecture model, the Roadmap to value model and the building block model.

1.4.5 Chapter 6: Continuous Architecture design

This chapter defines how Continuous Architecture provides direction for the realisation of the SOR, SOE and SOS.

1.4.6 Chapter 7: Application Continuous Architecture

This chapter describes how to apply architectural principles and architecture models in an Agile environment by describing the steps for each of the three value streams of Continuous Architecture.

1.4.7 Chapters 8 to 20

These chapters describe the architectural principles and architecture models of Continuous Everything value streams. The relationship with Continuous Architecture is also described.

1.4.8 Chapter 21

The maturity of Continuous Architecture has been made measurable in this chapter using a Continuous Architecture assessment.

1.5 Appendix

The appendices contain important information that will help you better understand Continuous Architecture.

Appendix	Subject	Explanation
A	Literature references	In this book reference is made to the literature consulted in the form of: [Auteur Jaar]. The appendix contains the author's full name, title, and ISBN number.
В	Glossary	Only the most important concepts are explained in this appendix.
С	Abbreviations	Many abbreviations are used within the world of DevOps. For the readability of this book, frequently used terms have been abbreviated. The first time an abbreviation is used, it is written out fully.
D	Websites	A number of relevant websites are included in this appendix. These websites are referred to in this book by the reference: [http Name].
E	Index	The index includes references to terms used in this book.

Table 1-2, Appendices.

1.6 Reading guidelines

The number of abbreviations has been kept limited in this book. However, terms that recur are shown as abbreviations to increase readability. Appendix C lists these abbreviations.

Appendices

Appendix A, Literature list

Table A-1 provides an overview of books that are directly or indirectly related to DevOps.

References	Publications
Best 2011a	B. de Best, "SLA best practice", Dutch language, Leonon Media 2011, ISBN13: 978 90 71501 456.
Best 2011b	B. de Best, "ICT Performance-Indicatoren", Dutch language, Leonon Media 2011, ISBN13: 978 90 71501 470.
Best 2012	B. de Best, "Quality Control & Assurance", Dutch language, Leonon Media 2012, ISBN13: 978 90 71501 531.
Best 2014a	B. de Best, "Acceptatiecriteria", Dutch language, Leonon Media, 2014, ISBN 13: 978 90 71501 784.
Best 2014c	B. de Best, "Cloud SLA, English language, Leonon Media, 2014 ISBN13: 978 90 9261 8009.
Best 2017a	B. de Best, "Beheren onder Architectuur", Dutch language, Leonon Media, 2017, ISBN13: 978 90 71501 913.
Best 2017c	B. de Best, "SLA Templates", English language, Leonon Media, 2017, ISBN13: 978 94 92618 030.
Best 2018a	B. de Best, "Agile Service Management with scrum", English language, Leonon Media, 2018, ISBN13: 978 94 9261 8085.
Best 2018b	B. de Best, "Agile Service Management with Scrum in Practice", English language, Leonon Media, 2018, ISBN13: 978 94 9261 8177.
Best 2018c	B. de Best, "DevOps best practice", English language, Leonon Media, 2018, ISBN13: 978 94 92618 078.
Best 2019	B. de Best, "DevOps Architecture", English language, Leonon Media, 2019, ISBN13: 978 90 71501 579.
Best 2021b	B. de Best, "Basiskennis IT", Dutch language, Leonon Media, 2021, ISBN13: 978 94 92618 573.
Best 2022 CA	B. de Best, "Continuous Auditing", English language, Leonon Media, 2022, ISBN13: 978 94 92618 757.
Best 2023 CC	B. de Best, "Continuous Acceptance", English language, Leonon Media, 2023, ISBN13: 978 94 91480 324.
Best 2022 CD	B. de Best, "Continuous Deployment", English language, Leonon Media, 2022, ISBN13: 978 94 92618 733.
Best 2024 CH	B. de Best, "Continuous Architecture", English language, Leonon Media, 2024, ISBN13: 978 94 91480 355
Best 2022 CI	B. de Best, "Continuous Integration", English language, Leonon Media, 2022, ISBN13: 978 94 92618 689.
Best 2022 CL	B. de Best, "Continuous Learning", English language, Leonon Media, 2022, ISBN13: 978 94 92618 740.
Best 2022 CM	B. de Best, "Continuous Monitoring", English language, Leonon Media, 2022, ISBN13: 978 94 92618 719.
Best 2022 CN	B. de Best, "Continuous Design", English language, Leonon Media, 2022, ISBN13: 978 94 92618 702.
Best 2022 CP	B. de Best, "Continuous Planning", English language, Leonon Media, 2022, ISBN13: 978 94 92618 726.
Best 2023 CQ	B. de Best, "Continuous SLA", English language, Leonon Media, 2023, ISBN13: 978 94 91480 256.
Best 2022 CS	B. de Best, "Continuous Assessment", English language, Leonon Media, 2022, ISBN13: 978 94 92618 696.

References	Publications
Best 2022 CT	B. de Best, "Continuous Testing", English language, Leonon Media, 2022, ISBN13: 978 94 92618 672.
Best 2022 CY	B. de Best, "Continuous Security", English language, Leonon Media, 2022, ISBN13: 978 94 91480 188.
Best 2023 CZ	B. de Best, "Continuous AI", English language, Leonon Media, 2023, ISBN13: 978 94 91480 300.
Best 2022a	B. de Best, "Continuous Development", English language, Leonon Media, 2022, ISBN13: 978 94 92618 764.
Best 2022b	B. de Best, "Continuous Operations", English language, Leonon Media, 2022, ISBN13: 978 94 92618 771.
Best 2022c	B. de Best, "Continuous Control", English language, Leonon Media, 2022, ISBN13: 978 94 91480 201.
Best 2022d	B. de Best, "Continuous Everything", English language, Leonon Media, 2022, ISBN13: 978 94 92618 665.
Bloom 1956	Benjamin S. Bloom, "Taxonomy of Educational Objectives (1956)", Allyn and Bacon, Boston, MA. Copyright (c) 1984 by Pearson Education.
Boehm 1981	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, 2016, ISBN-13: 978 14 91926 307.
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, 2015, ISBN-13: 978 14 91939 369.
Galbraith 1992	Galbraith, J.R. "Het ontwerpen van complexe organisaties", Alphen aan de Rijn: Samson Bedrijfsinformatie, 1992.
Humble 2010	Jez Humble, David Farley "Continuous Delivery Reliable Software Releases through Build, Test, and Deployment Automation", Addison- Wesley Professional; 1 edition, 2010, ISBN-13: 978 03 21601 919.
Kim 2014	Gene Kim, Kevin Behr, George Spafford "The Phoenix Project", IT Revolution Press, 2014, ISBN-13: 978 09 88262 508.
Kim 2016	Gene Kim, Jez Humble "The DevOps Handbook: How to Create World- Class Agility, Reliability, and Security in Technology Organisations, Patrick Debois, John Willis", 2016, IT Revolution Press, ISBN-13: 978 19 42788 003.
Kotter 2012	John P. Kotter, "Leading Change", Engels 1e druk, November 2012, ISBN13: 978 14 22186 435.
Kaplan 2004	R. S. Kaplan en D. P. Norton, "Op kop met de Balanced Scorecard", 2004, Harvard Business School Press, ISBN13: 978 90 25423 032.
Layton 2017	Mark C. Layton Rachele Maurer, "Agile Project Management for Dummies", tweede druk, John Wiley & Sons Inc, 2017, ISBN13: 978 11 19405 696.
Looijen 2011	M. Looijen, L. van Hemmen, "Beheer van Informatiesystemen", zevende druk, Academic Service, 2011, ISBN13: 978 90 12582 377.
MAES	R. Maes, "Visie op informatiemanagement", www.rikmaes.nl.
McCabe	McCabe T. "A Complexity Measure" in: IEEE Transactions on Software Engineering 1976, vol. 2, nr. 4.

References	Publications
Michael Porter 1998	M.E. Porter "acceptance criteria 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, "Informatiserings-economie", 2001, ISBN 90 395 1393 7.
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.

Table A-1, Literature list.

Appendix B, Glossary A glossary of terms is included in Table B-1.

Term	Meaning
55	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 (躾): 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 (<u>FAT</u>), User Acceptance Testcases (<u>UAT</u>) and Production Acceptance Testcases (<u>PAT</u>). The FAT and UAT should be expressed in the language of the business.
Affinity	DevOps is about <u>collaboration</u> 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 <u>happy path.</u>
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.
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> .

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

Term	Meaning
	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 "over the shoulder", pair programming, email pass-around and tool-assisted code review.
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: 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.

Term	Meaning
	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.
Compliancy officer	The compliancy officer is a DevOps role. The compliancy 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. Culture: 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.

Term	Meaning
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. 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].

Term	Meaning
Given-When-Then	The Given-When-Then format is used to define acceptance criteria in a way that the stakeholders understand how the functionality actually will work. GIVEN – the fact that WHEN – I do this THEN – this happens
Green field	See brown field.
Hand-off Readiness Review (HRR)	The HRR term is introduced by Google. An HRR is set of safety checks for a critical stage of releasing new services. HRR is performed when a service is transitioned from a developer-managed state to an OPS- managed state (usually months after the LRR). HRR makes service transition easier and more predictable and helps create empathy between upstream and downstream work centers.
Happy path	An application supports a business process by receiving, editing, storing, and providing information. The assumed steps in which the information processing is performed is called the happy path. The steps in alternate ways are called the alternate path. In that case, the same result will be achieved via another navigation path. The crawl of the application that causes an error is called an error path.
Holocracy	In this type of organisation all decisions are made through self- organising teams rather than through a traditional management hierarchy.
Horizontal splitting of features	A feature can be splitted into stories. Horizontal splitting refers to the result of a feature splitting in which more DevOps teams must work tightly together. They have to align their work continuously in order to deliver together the feature.
I-shaped, T-shaped, E-shaped	I-shaped, T-shaped, E-shaped are the categories to indicate the knowledge and special skills of a person. An I-shaped person is a pure specialist in one area. The T-shaped person has special skills in one field and broad general knowledge. The E-shaped person has special skills in more than one field and broad general knowledge.
Idempotent	Continuous Delivery requires that a component can always to be brought fully automatically to the desired status regardless of the component's initial state and regardless of the number of times the component is configured. The characteristic of a component to always be able to get back into the desires is called idempotent.
Imperative programming	This is a <u>programming paradigm</u> that uses statements that change a program's state. Imperative programming focuses on how a program should operate and consists of commands for the computer to perform. Examples are COBOL, C, BASIC et cetera. The term is often used in contrast to <u>declarative programming</u> , which
	focuses on what the program should accomplish without specifying how the program should achieve the result.
Independent, Negotiable, Valuable, Estimable, Small, and Testable (INVEST)	 Independent, Negotiable, Valuable, Estimable, Small, and Testable. 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.

Term	Meaning
	 Estimable: The size of a product backlog item must always estimable. Small: Product backlog items should not be so big as to become impossible to plan / task / prioritise with a certain level of certainty. Testable: The product backlog item or its related description must provide the necessary information to make test development possible.
Information radiators	An Information Radiator is a visual display that a team places in a highly visible location so that all team members can see the latest information at a glance.
Infosec	A team that is responsible for securing systems and data.
Infrastructure as Code (IaC)	Normally infrastructure components have to be configured in order to perform the requested functionality and quality for example a rule set for a firewall or the allowed IP addresses for a network. These configurations normally are stored in configuration files which enable the operators to manage the functionality and the quality of the infrastructure components. Infrastructure as code (IaC) makes it possible to programme these infrastructure component settings and deploy these settings through the CI/CD secure pipeline by the use of machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.
Infrastructure as Code (IaC)	Infrastructure as code (IaC) is a software-based approach to the ICT infrastructure, whereby the systems can be rolled out and adapted in a consistent manner through templates. If a change has to be made, it is implemented in the template which is then rolled out again.
Infrastructure management	Infrastructure management consists of the lifecycle management of all infrastructure products and services in order to support the correct working of the applications that run on top of the infrastructure.
Ji-Kotei-Kanketsu (JKK)	 JKK which means 100% completion of an item. This quality way of working means: clear understanding of the goals; understanding the right way to work; ensure high quality of work; getting the work right for 100% completion, never pass defects to the next process; Definition of Done (DoD) is vital; and then maintaining the required quality without inspections.
Just In Time (JIT)	JIT means building up a stream-lined supply chain with one-piece flow.
Kaizen	 Kaizen is Japanese for "improvement". Kaizen is used to improve production systems. The goals of kaizen are: elimination of waste (muda's); <u>JIT;</u> standardisation of production; cycle of continuous improvements.
	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. The following steps can be followed: • defining problems with supporting data; • making sure everybody recognises the problems clearly;

Term	Meaning
	 setting a hypothesis on the problems found; defining countermeasure actions to verify the hypothesis; defining countermeasure actions be in daily based activities; measuring a weekly KPI so people can feel a sense of accomplishment.
Kaizen Blitz (or Improvement Blitz)	A Kaizen Blitz is a rapid improvement workshop designed to produce results / approaches to discrete process issues within a few days. It is a way for teams to carry out structured, but creative problem solving and process improvement, in a workshop environment, over a short timescale.
Kaizen in advance	Kaizen in advance goes one step further than Kaizen. Not only the own activities are improved but also the activities that are performed upstream and that lead to problems downstream. In this way a feedback loop of problems is created which improves the system as a whole.
Kanban	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.
	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.
Kata	 A kata is any structured way of thinking and acting (pattern of behavior) that is practiced until the pattern becomes a second nature. Four steps can be recognised to accomplish this second nature: direction (target); current condition (IST situation); target condition (SOLL situation); PDCA (Deming wheel). 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.
Kibana dashboards	A Kibana dashboard displays a collection of saved visualisations.
Latent defects	Problems that are not visible yet. Latent defects can be made visible by injecting faults into the system.
Launch Readiness Review (LRR)	The LRR term is introduced by Google. An LRR is a set of safety checks for a critical stage of releasing new services. It is performed and signed off before a service is made publicly available and receive live production traffic. LRR is self-reported by the project teams. LRR is used in the development-managed state.
Launching guidance	To prevent the possibility of problematic, self-managed services going into production and creating organisational risk, launch requirements may be defined that must be met in order for services to interact with real customers and be exposed to real production traffic [Kim 2016].
Lead Time (LT)	Lead time is the time from when a request is made to when the final result is delivered, or the customer's point of view on how long something takes to complete.
Lean tools	A3 thinking (problem solving)Continuous flow (eliminates waste)

Term	Meaning
	 <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	A learning culture is a collection of organisational conventions, values, practices, and processes. These conventions encourage employees and organisations to develop knowledge and competence. 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.
Light weight ITSM	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.
Logging levels	 Within monitoring systems there are several levels of logging recognised: 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	Loosely coupled architectures enables that changes can be made safely and with more autonomy, increasing developer productivity.
Micro service	Microservices are a variant of the service-oriented architecture (SOA) architectural style that structures an application as a collection of loosely coupled services. In a microservices architecture, services should be fine-grained, and the protocols should be lightweight [Wiki].
Micro service architecture	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.
Mini pipeline	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. 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.

Term	Meaning
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: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: 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: 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.

Term	Meaning
	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 a 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]: Environment-based release patterns: In this pattern there are two or more environments that receive deployments, but only one environment is receiving live customer traffic. Application-based release patterns: In this pattern the application is modified in order to make selectively releases possible and to expose specific application functionality by small configuration changes.
Sad path	A specific type of a ' <u>bad path</u> ' is called a 'sad path'. This is the case if the 'bad path' results in a security-related error condition.
Safety checks	Safety checks are performed during a release of a product. They are typical part of an <u>HRR</u> of an <u>LRR</u> .

Term	Meaning
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: 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: Janitor Monkey (unused resources); Chaos Monkey (try to shut down a service); Conformity Monkey (non-conformance to rules).
Single repository	A single repository is used to facilitate trunk-based development.
Smoke testing	Smoke testing is one of the test types that is used to determine whether or not the basics of a new or adjusted service works. Only a few testcases are needed to indicate whether or not at least the most important functions are working properly. This test type origins from the hardware manufacturers where engineers tested circuits by powering on the system and checking for smoke which was an alarm of malfunctioning hardware.
Standard deviation	In statistics, the standard deviation (SD, also represented by the Greek letter sigma σ or the Latin letter s) is a measure that is used to quantify the amount of variation or dispersion of a set of data values. A low standard deviation indicates that the data points tend to be close to the mean (also called the expected value) of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values [Wiki].
Standard operations	The standard operations is the situation in which the system performs as designed. Deviations of the standard operations need to be detected as early as possible.

Term	Meaning
Static analysis	Static analysis is a type of testing that is performed in a non-runtime environment, ideally in the deployment pipeline. Typically, a static analysis tool will inspect program code for all possible run-time behaviours and seek out coding flaws, back doors, and potentially malicious code [Kim 2016].
Swarming	David Bernstein explains how swarming helps to build an effective team which is able to focus and solve complex problems: "When swarming, the whole team works together on the same problem. It helps to know each other and work well together. Generally, groups need to go through the phases of forming (getting to know each other) and storming (having conflicts and resolving them) before they get to performing (being a highly functional team), so give everyone the space to become a team.".
	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].
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)	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.
	To ensure data integrity, there must be one and only one system of record for a given piece of information [What-is].
Technology adaption curve	It takes time for new technology to get adapted in the market. The technology adaption curve indicates the stages of market penetration in time.
Technology executives	This is a DevOps role also named 'value stream manager'. The value stream manager is someone who is responsible for "ensuring that the value stream meets or exceeds the customer (and organisational) requirements for the overall value stream, from start to finish" [Kim 2016].
Test Driven Development (TDD)	Test driven development is the approach in which the source code is written after the completion of the test case definition and execution. The source code is written and adjusted until the test case conditions are met.
Test harness	Software constructed to facilitate integration testing. Where test stubs are typically components of the application under development and are replaced by working components as the application is developed (top-down integration testing), test harnesses are external to the application being tested and simulate services or functionality not available in a test environment.
The Agile Manifesto	The Agile Manifesto (Manifesto for Agile Software Development) was set up during an informal meeting of seventeen software DevOps engineers. This meeting took place from 11 to 13 February 2001 at "The Lodge" in Snowbird, Utah.

Term	Meaning
The ideal testing automation pyramid	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. After the publication of the charter, several signatories set up the "Agile Alliance" to further convert the principles into methods [Wiki]. The ideal testing automation pyramid is a way of testing that can be characterised as follows:
	 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	 The non-ideal testing automation pyramid is a way of testing that can be characterised as follows: 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	The Simian Army is a collection of open-source cloud testing tools created by the online video streaming company, Netflix. The tools allow engineers to test the reliability, security, resiliency and recoverability of the cloud services that Netflix runs on Amazon Web Services (AWS) infrastructure [Whatis]. Within this Simian Army the following monkeys are recognised: Chaos Gorilla, Chaos Kong, Conformity Monkey, Doctor Monkey, Janitor Monkey, Latency Monkey and Security Monkey.
The three ways	 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. The first way – flow understand and increase the flow of work (left to right); The second way – feedback create short feedback loops that enable continuous improvement (right to left); The third way – Continuous Experimentation and Learning (continuous learning).
Theory of constraints	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].

Term	Meaning
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: Unnecessary software features. Communication delays. Slow application response times. Overbearing bureaucratic processes.
Waste reduction	Minimisation of waste at its source is to minimise the quantity required to be treated and disposed of, achieved usually through better product design and/or process management. Also called waste minimisation [Businessdictionary].
WIP limit	This is a Key Performance Indicator (KPI) that is used in the Kanban process to maximise the number of items that has been started but that is not completed. Limiting the amount of WIP is an excellent way to increase throughput in your software development pipeline.
Work In Progress (WIP)	Material that has entered the production process but is not yet a finished product.

216 DevOps Continuous Architecture

Term	Meaning
	Work in progress (WIP) therefore refers to all materials and partly finished products that are at various stages of the production process.

Table B-1, Glossary.

Appendix C, Abbreviations

Abbreviation	Meaning
%C/A	Percent Complete / Accurate
ASL	Application Services Library
AWS	Amazon Web Services
BDD	Behavior Driven Development
BI	Business Intelligence
BiSL	Business Information Services Library
ВОК	Body of Knowledge
BSC	Balanced Scorecard
BVS	Business Value System
СА	Competitive Advantage
СА	Continuous Auditing
САВ	Change Advisory Board
CAMS	Culture, Automation, Measurement and Sharing
CD	Continuous Deployment
CE	Continuous Everything
СЕМ	Central Event Monitor
CEMLI	Configuration, Extension, Modification, Localisation, Integration
CEO	Chief Executive Officer
CFO	Chief Finance Officer
CI	Configuration Item
CI	Continuous Integration
CIA	Confidentiality, Integrity & Accessibility (or Availability)
CIO	Chief Information Officer
CL	Continuous Learning
СМ	Continuous Monitoring
СМДВ	Configuration Management DataBase
СММІ	Capability Maturity Model Integration
CMS	Configuration Management System
CN	Continuous desigN
СО	Continuous dOcumentation
CoC	Code of Conduct
СоР	Communities of Practice
СР	Continuous Planning
CQ	Continuous SLA
CPU	Central Processing Unit
CR	Competitive Response
CRAMM	CCTA Risk Assessment Method Methodology
CRC	Cyclic Redundancy Check
CS	Continuous aSsessment
CSF	Critical Success Factor
СТ	Continuous Testing
СТО	Chief Technical Officer

Abbreviation	Meaning	
CY	Continuous securitY	
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 Bus	
ETL	Extract Transform & Load	
EUX	End User eXperience Monitoring	
FAT	Functional Acceptance Test	
FSA	Feature Solution Approach	
GCC	General Computer Controls	
GDPR	General Data Protection Regulation	
GIT	Global Information Tracker	
GSA	Generic & Specific Acceptatiecriteria	
GUI	Graphical User Interface	
GWT	Given-When-Then	
HRM	Human Resource Management	
HRR	Hand-off Readiness Review	
IaC	Infrastructure as Code	
ICT	Information Communication Technology	
ID	Identifier	
INVEST	Independent, Negotiable, Valuable, Estimatable, Small and Testable	
IPOPS	Information assets, People, Organisation, Products, and services, Systems, and processes	
IR	Infrastructure Risk	
ISAE	International Standard On Assurance Engagements	
ISMS	Information Security Management System	
ISO	Information Standardisation Organisation	
ISVS	Information Security Value System	
IT	Information Technology	
ITIL	Information Technology Infrastructure Library	
ITSM	Information Technology Service Management	
JIT	Just In Time	
ЈКК	Ji-Kotei-Kanketsu	
JVM	Java Virtual Machine	

Abbreviation	Meaning	
KPI	Key Performance Indicator	
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	
NLP	Natural Language Programming	
OAWOW	One Agile Way of Working	
OLA	Operational Level Agreement	
PAAS	Platform As A Service	
РАТ	Production Acceptance Test	
PBI	Productie Backlog Item	
PDCA	Plan Do Check Act	
PESTLE	Political, Economic, Sociological, Technological, Legislative, Environmental	
POR	Project or Organisational Risk	
РРТ	People, Process & Technology	
PST	Performance StressTest	
РТ	Processing Time	
QA	Quality Assurance	
QC	Quality Control	
RACI	Responsibility, Accountable, Consulted, and Informed	
RASCI	Responsibility, Accountable, Supporting, Consulted and Informed	
RBAC	Role Based Access Control	
REST API	REpresentational State Transfer Application Programming Interface	
RL	Reinforcement Learning	
ROI	Return On Investment	
RUM	Real User Monitoring	
S-CI	Software Configuration Item	
SA	Strategic IS Architecture	
SAFe	Scaled Agile Framework	

Abbreviation	Meaning	
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	System Integration test	
SLA	Service Level Agreement	
SM	Strategic Match	
SMART	Specific, Measurable, Accountable, Realistic, Timely	
SME	Subject Matter Expert	
SNMP	Simple Network Management Protocol	
SoA	Statement of Applicability	
SoE	System of Engagement	
SoI	Systems of Information	
SoR	System of Records	
SoX	Sarbanes Oxley	
SQL	Structured Query Language	
SRG	Standards Rules & Guidelines	
SSL	Secure Sockets Layer	
ST	System test	
SVS	Service Value System	
SWOT	Strength, Weakness, Opportunities, Threats	
ТСО	Total Cost of Ownership	
ТСР	Transmission Control Protocol	
TDD	Test Driven Development	
TFS	Team Foundation Server	
TISO	Technical Information Security Officer	
ТОМ	Target Operating Model	
TPS	Toyota Production System	
ТТМ	Time To Market	
TU	Technical Uncertainty	
UAT	User Acceptance Test	
UML	Unified Modeling Language	
UT	Unit Testing	
UX design	User eXperience Design	
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	

Abbreviation	Meaning
XP	eXtreme Programming
Table C-1, Abbreviations.	

Appendix D, Websites

bigpanda	[Bigpanda]	https://www.bigpanda.io/blog/event- correlation/
Bullseye	[Bullseye]	https://www.bullseye.com/minimum.html
Businessdictionary	[Businessdictionary]	http://www.businessdictionary.com
Collabnet	[CollabNet]	https://www.collab.net
CleanArchitecture	[CleanArchitecture]	https://www.freecodecamp.org/news/a-quick- introduction-to-clean-architecture- 990c014448d2/
CleanCode	[CleanCode]	https://cvuorinen.net/2014/04/what-is-clean- code-and-why-should-you-care/
dbmetrics	[dbmetrics]	http://www.dbmetrics.nl
dbmetrics	[dbmetrics publicaties]	https://www.dbmetrics.nl/wp- content/uploads/2021/07/dbmetrics_best- practice-publications_2021-07-22_900.pdf
De Caluwé	[De Caluwé]	https://www.agile4all.nl/het-kleurenmodel-van- de-caluwe-en-vermaak/
DevOps	[DevOps]	http://DevOps.com
DDD	[DDD]	https://www.slideshare.net/skillsmatter/ddd-in- agile
doxygen	[doxygen]	http://www.doxygen.nl/manual/docblocks.html
doxygen example	[doxygen example]	http://www.doxygen.nl/manual/examples/qtstyl e/html/class q tstyle test.html#a0525f798c da415a94fedeceb806d2c49
EXIN	[Exin]	http://www.exin.nl
Gladwell	[GLADWELL]	http://www.gladwill.nl
IIR	[IIR]	http://www.IIR.nl
Investopedia	[Investopedia]	https://www.investopedia.com
ITMG	[ITMG]	http://www.ITMG.nl
ITPedia	[ITPEDIA]	http://www.itpedia.nl
Patrick Cousot	[Patrick Cousot]	https://www.di.ens.fr/~cousot/abstract_interpr et.shtml
Porter	[Porter]	https://medium.com/@sniloy/value-chain- analysis-value-stream-mapping-and-business- process-mapping-what-is-the-difference- 431589d27ea8
Sneider	[Schneider]	https://shift314.com/are-you-using-the-right- culture-model/
Tiobe	[Tiobe]	www.tiobe.com/content/paperinfo/DefinitionOfC onfidenceFactor.html
UnitTest	[UnitTest]	https://docs.python.org/3/library/unit test.html
Westrum	[Westrum]	https://www.delta-n.nl/het-belang-van-cultuur- in-devops/
Wiki	[Wiki]	http://nl.wikipedia.org / wiki / Cloud_computing
Wiki docgen	[Wiki docgen]	https://en.wikipedia.org/wiki/Comparison_of_do cumentation_generators

Table D-1, Websites.

Appendix E, Index

%

%C/A · 217

A

A/B testing · 201 acceptance - criteria · 85, 202, 206 - environment · 131 - test · 201 actor · 32, 34, 36, 39 affinity · 201 Agile · 201, 213, 214 Agile infrastructure · 201 Agile Scrum · 205 Agile Scrum framework · 18 AI application · 118, 120, 121, 122 AI pattern library · 123 alternate path · 201 Amazon Web Services · See AWS Andon cord · 201 anomaly detection technique · 201 antifragility · 202 anti-pattern · 95, 201 application component · 209 application management · 208 Application portfolio management · 50 Application Services Library · See ASL Archimate plate · 54, 78, 83, 165, 176 architectural knowledge · 22 architecture · 2, 7, 74 artefact · 202, 204 artefact repository · 202 ASL · 217 assessment · 212 Atomic operation · 114 Auditing Pyramid model \cdot 165 authorities · 63 automated test · 202 availability · 203 AWS · 28, 53, 92, 143, 214, 217 Azure · 53, 92, 143

B

backlog item · 211 bad apple theory · 202 bad path · 202 Balanced Score Card · See BSC BDD · 64, 81, 101, 136, 202, 217 Behavior Driven Development · See BDD best practice · 203 besturingsmodel · 23, 61, 84, 118, 141, 146 BI · 217 binary · 202 BiSL · 217 blameless post mortem · 202 blamelessness · 202 Bloom · 149 Bloom level · 149 blue/green deployment · 202 Body of Knowledge · See BOK BOK · 150, 217 bottleneck · 19, 26, 56, 58, 66, 70, 76, 83, 84, 88, 117, 134, 151, 152 boundary · 24, 38, 39, 55, 56, 58, 63, 67, 75, 79, 84, 88, 135, 187 branching · 203 broken build · 202 brown field · 202 BSC · 5, 12, 26, 41, 42, 43, 45, 62, 66, 69, 74, 75, 77, 146, 160, 166, 175, 189, 193, 194, 217 build · 202, 203, 204, 213 building block · 25, 27, 28, 48, 49, 50, 51, 52, 54, 69, 75, 76, 83, 84, 88, 89, 90, 91, 92, 95, 96, 106, 160, 184, 185, 189, 193 business - case · 20 - hypothesis · 57 - model canvas · 41, 43, 44, 77, 166 - rule · 49, 89, 90 - service monitoring · 139 - strategy · 2, 34, 42, 193 - value · 203, 205 Business Information Services Library · See BiSL Business Intelligence · See BI Business Value System · See BVS BVS · 5, 45, 217

С

CA · 217 CAB · 217 CAMS · 204, 217 Canary release pattern · 131 canary releasing · 203 capability · 204 Capability Maturity Model Integration · See CMMI capacity · 203 CCTA Risk Assessment Method Methodology · See CRAMM CD · 161, 190, 203, 207, 217 CE · 217 - maturity model · 160 - model · 189 value stream · 193, 194 CEM · 217 CEMLI · 217

CE-model · 189 Central Event Monitor · See CEM Central Processing Unit · See CPU CEO · 217 CFO · 217 Change Advisory Board · See CAB change category · 203 change schedule · 203 change vision \cdot 56 Chief Executive Officer · See CEO Chief Finance Officer · See CFO Chief Information Officer · See CIO Chief Technology Officer · See CTO CI · 161, 189, 190, 203, 207, 217 CI/CD pipeline · 193 CI/CD secure pipeline · 11, 18, 39, 75, 86, 96, 98, 99, 125, 126, 128, 152, 192 CIA · 217 CIO · 217 CL · 161, 190, 194, 214, 217 cloud · 203 cloud configuration file · 203 cloud service · 203 cluster immune system release pattern · 203 CM · 161, 190, 217 CMDB · 217 CMMI · 162, 191, 217 CMS · 217 CN · 217 CO · 217 CoC · 217 code branch · 203 code loss · 115 Code of Conduct · See CoC code review form · 203 codified NFR · 203 collaboration · 203 commit code · 203 commit stage · 203 Communities of Practice · See CoP competence · 201, 206 Competitive Advantage · See CA Competitive Response · See CR Completeness / Accurateness · See %C/A compliance · 204 compliance checking · 204 compliancy · 204 compliancy officer · 204 component · 207, 210, 213 Confidentiality, Integrity & Accessibility · See CIA configuration error · 115 Configuration Item · See CI configuration management · 204 Configuration Management DataBase · See CMDB Configuration Management System · See CMS Configuration, Extention, Modification, Localisation, Integration · See CEMLI container · 204 container service · 54, 93

continuity · 203, 209 Continuous - Acceptance · 83 - AI · 117 - Architecture · 11 - Assessment · 2, 32, 35, 38, 155 - Auditing \cdot 2, 165 - Auditing pyramid · 169, 170, 171 - Deployment · 2, 125 - Design · 2, 77 - Design pyramid · 77, 80, 81, 170 - Everything · 161, 162, 190, 191 - Improvement · 214 - Integration · 2, 105 - Learning · 2, 145, 159 - Monitoring · 2, 133 - Monitoring Layer model · 140 - Planning · 2, 33, 36, 38, 61 - Planning model · 61, 65, 66 - Security · 175 - Security control model · 182 - Security pyramid · 179, 180, 181 - SLA · 69 - Testing · 2, 95 - Testing roadmap · 13 Continuous aSsessment · See CS Continuous Auditing · See CA Continuous Deployment · See CD Continuous desigN · See CN Continuous dOcumentation · See CO Continuous Everything · See CE Continuous Integration · See CI Continuous Learning · See CL Continuous Monitoring · See CM Continuous Planning · 217, See CP Continuous securitY · See CY Continuous SLA · 217, See CQ Continuous Testing · See CT control · 204, 212 Conway's law · 204 CoP · 19, 217 countermeasure · 69, 71, 85, 208 coverage ration \cdot 20, 21 CP · 217 CPU · 217 CPU consumption · 108 CPU reduction · 108 CQ · 217 CR · 217 CRAMM · 217 CRC · 217 Critical Success Factor · See CSF CRM · 49, 90 CS · 217 CSF · 217 CT · 161, 190, 217 CTO · 217 cube assessment · 158 cube model · 155 cultural debt · 204 Culture, Automation, Measurement and Sharing \cdot See CAMS

current state · 12, 48, 50, 53, 54, 55, 88

CVS · 115 CY · 218 cycle time · 204 Cyclic Redundancy Check · See CRC

D

dark launching pattern · 132 datakwaliteit · 86, 122 debt · 204 declarative programming · 204 Deep Learning · See DL defect · 209 defect tracking · 205 Definition of Done · See DoD Definition of Ready · See DoR Definitional Uncertainty · See DU Definitive Media Library · See DML Demming wheel · 208 deployment · 201 deployment pipeline · 203 design · 204, 215, 220 development · 201, 202, 205, 207, 208, 210, 212, 213, 214, 215 Development & Operations · See DevOps development ritual · 205 Development Value System · See DVS Development, Test, Acceptance and Production · See DTAP DevOps · 170, 189, 191, 197, 201, 203, 205, 211, 215, 218 - engineer · 22 - Lemniscate · 180 - team · 201, 202, 203, 204, 206, 210, 215 DevOps engineer · 1, 2, 12, 19, 22, 24, 26, 36, 39, 63, 66, 84, 96, 97, 98, 106, 107, 108, 109, 110, 111, 112, 114, 115, 116, 126, 127, 129, 135, 137, 145, 147, 156, 157, 158, 201, 203, 204, 205, 212, 213, 214 DL · 218 DML · 218 DMZ service \cdot 54, 93 DNS · 54, 93, 218 Docker Enterprise · 54, 93 DoD · 33, 34, 36, 38, 135, 207, 218 Domain Name System · See DNS DoR · 218 downward spiral · 205 DTAP · 212, 218 DTAP environment · 212 DTAP street · 23, 62, 78, 96, 131, 134, 146, 156, 166, 176 DU · 218 DVS · 5, 45, 72, 218

Ε

E2E \cdot 159, 218 e-mail pass around \cdot 205

End User eXperience Monitoring · See EUX End-to-End · See E2E enterprise architecture · 7, 8, 17, 20, 26, 47 Enterprise Resource Planning · See ERP enterprise roadmap · 61 Enterprise Service Bus · See ESB Entity Relation Diagram · See ERD epic · 8, 57, 58, 218 epic one pager \cdot 58 epic owner \cdot 57 Epic Solution Approach · See ESA ERD · 49, 218 ERP · 18, 49, 90, 218 error path · 205 ESA · 218 ESB · 218 E-shaped · 206 E-shaped pattern · 152 E-shaper · 96, 107, 126, 147, 150 Esperanto · 25 ETL · 52, 92, 218 EUX · 139, 142, 218 event · 138, 210 event catalogue · 85 event log \cdot 51, 92 eXtensible Markup Language · See XML Extract Transform & Load · See ETL eXtreme Programming · See XP

F

failure · 202 fast feedback · 95 FAT · 97, 98, 99, 100, 101, 102, 103, 104, 201, 218 feature · 8, 65, 205, 206, 215 feature flag pattern · 131 Feature Solution Approach · See FSA feature toggle · 205 feedback · 159, 204, 205, 208, 211, 214 feedforward · 205 financial indicator · 42 flow · 204, 207, 208, 209, 210, 211, 212, 214, 215 forward release pattern · 132 framework · 210 FSA · 218 FTP · 54, 93 full release pattern · 131 Functional Acceptance Test · See FAT future state · 5, 12, 25, 48, 49, 50, 53, 54, 74,88

G

Gaussian distribution · 201, 205 GCC · 218 GDPR · 86, 218 Gene Kim · 205, 210, 214 General Computer Controls · See GCC

General Data Protection Regulation · See GDPR Generic & Specific Acceptatiecriteria · See GSA Gherkin language · 64, 79, 136, 167, 177 GIT · 115, 218 Given When Then \cdot 206, See GWT Global Information Tracker · See GIT qoal · 32, 33, 36, 38 governance model · 138 governance structure · 25, 63, 64, 120, 148, 168, 178 Graphical User Interface · See GUI green field · 206 GSA · 218 GUI · 103, 218 guiding coalition · 56 GWT · 65, 81, 99, 206, 218

Η

Hand-off Readiness Review · See HRR happy path · 201, 202, 206 hardware · 204, 207, 215 high level requirement · 24, 78, 83, 95, 105, 117, 125, 134, 165, 176, 193 high performance model · 125, 133, 145, 155, 165, 175 high-level requirement · 12, 24, 36, 56, 84, 96, 106, 118, 125, 176 holocracy · 206 horizontal splitting of feature · 206, 215 HRM · 22, 23, 42, 62, 70, 78, 134, 135, 147, 149, 157, 166, 176, 218 HRR · 218 Human Resource Management · See HRM

I

- IaC · 54, 93, 103, 201, 207, 218 ICT · 207, 218 ID · 218 ideal test pyramid · 99, 100, 161, 190, 214 idempodent · 206 IDentifier · See ID imparative programming · 206 incremental release pattern · 131 Independent, Negotiable, Valuable, Estimatable, Small and Testable · See INVEST Information assets, People, Organisation, Products and services, Systems and processes · See IPOPS Information Communication Technology · See ICT information radiator · 207 Information Security Management System · See ISMS
- Information Security Value System · See ISVS

Information Standardisation Organisation · See ISO Information Technology · See IT Information Technology Infrastructure Library · See ITIL Information Technology Service Management · See ITSM Infosec · 207 Infrastructure as Code · See IaC infrastructure component · 207 infrastructure management · 207 Infrastructure Risk · See IR integrity rule \cdot 49, 90, 120 International Standard On Assurance Engagements · See ISAE INVEST · 206, 218 IP address · 207 IPOPS · 218 IR · 218 ISAE · 218 I-shaped · 206 ISMS · 218 ISO · 218 ISO 27001 · 45, 86, 183, 184, 187 IST · 208 ISVS · 5, 45, 218 IT · 205, 209, 214, 218 ITIL · 45, 46, 52, 92, 186, 218 ITIL 4 · 45, 46, 186 ITSM · 209, 218

J

Java Virtual Machine · See JVM Ji-Kotei-Kanketsu · See, See JKK JIT · 207, 208, 218 JKK · 207, 218 Just In Time · See JIT JVM · 218

K

Kaizen · 207, 209 Kaizen Blitz (or Improvement Blitz) · 208 Kaizen in advance · 208 Kanban · 208, 209, 215 Kaplan · 42, 198 Key Performance Indicator · See KPI kibana dashboard · 208 knowledge · 70, 206, 209, 213, 233 KPI · 57, 208, 209, 215, 219

L

LAN · 219 Laravel · 52, 92, 103, 104 latent defect · 208 Launch Readiness Review · See LRR launching guidance · 208

laws & regulations · 86, 122 Layton · 57 LCM · 219 LDAP · 219 LDAP service · 54, 93 Lead Time · 208, See LT Lean · 214, 215 Lean indicator · 9, 46, 56, 75, 87, 88, 138, 139 Lean tool · 208 learning culture · 209 Lemniscate · 1, 2, 155, 170, 180 lifecycle · 205, 207 LifeCycle Management · See LCM Lightweight Directory Access Protocol · See LDAP limitation · 20, 56, 75, 79, 84, 135, 138 Local Area Network · See LAN local file system · 110, 111 $\log \cdot 210$ logging level · 209 long lived branch \cdot 114 Looijen \cdot 45, 198 loosely coupled \cdot 20, 21 loosely coupled architecture · 209 loosely coupled service · 209 LRR · 208, 219 LT · 208, 219

Μ

Machine Learning · See ML machtsverhouding · 18 Management Information · See MI manufacturing process · 215 marker · 36, 38 MASR · 219 master data · 49, 51, 89, 91 Mean Time Between Failure · See MTBF Mean Time Between System Incidents · See MTBSI Mean Time To Repair · See MTTR merge hell · 115 metadata · 85, 97, 98, 107, 121, 127, 128, 129, 130, 202 MFA · 219 MI · 219 Michael Porter · 9, 44, 45, 46 microservice · 209 microservice architecture · 209 Microsoft Operations Framework · See MOF migration path · 2, 6, 7, 12, 25, 74, 97, 107, 127, 160, 189, 208 mini pipeline · 209 Minimal Viable Product · See MVP Minimum Required Information · See MRI ML · 219 Modify, Avoid, Share, Retain · See MASR module \cdot 103 Module Test · See MT MOF · 219 monitoring · 210, 217

Monitoring Layer model · 133, 138, 139, 140 monolithic · 210 MRI · 209, 219 MT · 219 MTBF · 219 MTBSI · 219 MTTR · 210, 219 muda · 210 Multi Factor Authentication · See MFA MVP · 8, 57, 193, 219

Ν

Natural Language Programming · See NLP NC · 219 Necker cube · 158 NFR · 58, 203, 210, 219 NLP · 219 Non Conformity · See NC Non Functional Requirement · See NFR

0

OAWOW · 219 obeya · 210 object code · 202 OLA · 219 One Agile Way of Working · See OAWOW one piece flow · 210 open source · 114 operating system \cdot 115 Operational Level Agreement · See OLA operations · 201, 205, 210, 212, 215 operations story · 210 Ops liaison · 210 organisation archetype · 210 organisation strategy \cdot 63 organisational typology model · 210 OSI model · 54, 93 outcome · 43 over-the-shoulder · 211

Ρ

PAAS \cdot package \cdot pair programming \cdot 203, 211 PAT \cdot 75, 99, 100, 101, 102, 103, 104, 201, 219 pattern \cdot 131, 201, 202, 203, 208, 211 \cdot blue / green deployment \cdot \cdot canary release \cdot \cdot cluster immune system \cdot \cdot container pattern \cdot \cdot dark launching \cdot \cdot feature flag \cdot \cdot forward release \cdot \cdot full release \cdot

- incremental · 131 PBI · 23, 100, 101, 102, 103, 104, 219 PDCA · 208, 209, 219 peer review · 211 peer to peer programming · 203 People, Process & Technology · See PPT PEP · 22, 147 performance · 32, 34, 36, 39, 203, 209, 215, 219 Performance StressTest · See PST PESTLE · 219 PHP framework · 52, 92, 103, 104 PI · 19 pipeline · 201, 207, 209, 212, 213, 215 Plan Do Check Act · See PDCA Planning & Design model · 65, 66, 67 Platform As A Service · See PAAS Plug-in IDE · 114 Political, Economic, Sociological, Technological, Legislative, Environmental See PESTLE POR · 219 Porter · 45, 46, 199, 223 portfolio · 17, 18, 33, 34, 44, 47, 48, 49, 50, 52, 54, 61, 69, 74, 78, 121, 122, 138, 146, 165, 176, 193 post mortem · 211 PPT · 23, 62, 70, 71, 78, 134, 146, 156, 166, 176, 219 Processing Time · See PT product - backlog · 12, 13, 23, 36, 37, 56, 62, 63, 64, 66, 69, 71, 72, 75, 77, 81, 88, 100, 101, 106, 118, 147, 157, 160, 189, 206, 207, 211 - backlog item · 207 - owner · 57, 63, 211 - roadmap · 6, 12, 18, 20, 27, 36, 37, 38, 62, 65, 73, 75, 77, 106 - vision · 26, 36, 65, 73 Product Backlog Item · See PBI Production AcceptatieTest · See PAT production environment · 202, 209, 210 programming paradigm · 211 Project or Organisational Risk · See POR PSQL · 204 PST · 99, 100, 101, 102, 103, 104, 219 PT · 219 pull request process · 211 **PVCS** · 115

Q

QA · 104, 155, 158, 159, 161, 190, 211, 219 QC · 219 Quality Assurance · See QA Quality Control · See QC

R

RACI · 219 RASCI · 19, 20, 24, 63, 71, 72, 78, 84, 97, 107, 118, 127, 135, 147, 158, 166, 176, 219 RBAC · 219 RBAC service · 54, 93 Real User Monitoring · See RUM reduce batch size · 211 reduce number of handoffs · 211 Reinforcement Learning · See RL release · 2, 211 release manager \cdot 211 release pattern · 211 repository · 202, 203, 211, 212 **REpresentational State Transfer Application** Programming Interface · See REST API requirement · 202, 208, 210, 213, 219 Responsibility, Accountable, Consulted and Informed · See RACI Responsibility, Accountable, Supporting, Consulted and Informed · See RASCI REST API · 50, 54, 91, 93, 140 REST-API · 103, 219 retrospective · 205 Return On Investment · See ROI review · 203, 205, 211, 214 risk · 115, 202, 208 risk based planning · 21, 64 risk management · 1, 72, 83, 100, 179, 185 RL · 219 roadmap · 5, 6, 8, 12, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25, 27, 33, 35, 37, 38, 39, 40, 50, 56, 57, 59, 61, 63, 64, 65, 66, 69, 70, 72, 73, 75, 78, 97, 105, 107, 118, 121, 127, 129, 130, 145, 147, 148, 150, 160, 167, 168, 177, 178, 189, 193 roadmap to value · 26, 27, 65, 73 Roadmap to value model \cdot 2, 5, 6, 26, 65, 73 ROI · 219 Role-based access control · See RBAC root cause · 207 root cause analysis · 209 RUM · 139, 219

S

SA · 219 sad path · 211 SAFe · 219 safety check · 211 Sarbanes Oxley · See SoX SAT · 100, 101, 102, 103, 104, 220 SBAR · 212, 220 SBB · 33, 61, 62, 88, 92, 220 SBB diagram · 27, 29 SBB layer · 27, 88 SBB type · 27, 88 SBB-A · 220 SBB-I · 220 SBB-T · 32, 33, 220 SBB-T diagram · 53 SBI · 71, 72 Scaled Agile Framework · See SAFe S-CI · 219 Secure Sockets Layer · See SSL security · 203, 210, 211, 212, 214 Security Acceptance Test · See SAT security control · 179 security officer · 203 self service capability · 212 sense of urgency \cdot 56 service · 219 Service Level Agreement · See SLA Service Value System · See SVS shared goals · 212 shift left organisation · 15, 97, 98, 107, 127 silo · 215 Simian army · 212, 214 Simple Network Management Protocol · See SNMP SIP · 71 SIT · 100, 101, 102, 103, 104, 220 Situation, Background, Assessment, Recommendation · See SBAR skill · 206 SLA · 197, 220 SLA control · 56, 69, 71, 72, 75 SLA norm · 71, 72, 85, 136, 141 SM · 220 SMART · 209, 220 SMART goal · 26, 66, 74 SME · 220 smoke testing \cdot 212 SNMP · 220 SOA · 220 SOE · 3, 213, 220 SOE application · 11, 12, 16, 17, 23, 26, 56, 58, 117 software \cdot 101, 202, 213, 215 Software Configuration Item · See S-CI SOI · 213, 220 SOLL · 208 Solution architect · 39, 40 SOR · 3, 5, 213, 220 SOR application · 5, 6, 11, 12, 13, 16, 17, 18, 19, 20, 23, 25, 26, 31, 33, 34, 35, 36, 41, 56, 58, 59, 61, 67, 69, 77, 106, 117, 118, 134, 166, 192, 193 SOS · 3, 5, 7, 11, 12, 16, 17, 18, 25, 31, 37, 38, 40, 58, 59, 61, 67, 69, 77, 83, 95, 105, 117, 118, 125, 134, 146, 156, 165, 175, 192, 193 SOS service · 11 source code · 202, 203, 205, 212, 213, 214 SoX · 220 Specific, Measurable, Accountable, Realistic, Timely · See SMART SPOF · 24

Spotify model · 19 sprint · 8, 205 - backlog · 24, 100, 101 - execution · 205 - planning · 205 SQL · 220 SRG · 220 SSL · 220 ST · 220 stakeholder · 73, 206 standard deviation · 212 standard operations · 212 Standard Rules & Guidelines · See SRG stand-up · 205 Statement of Applicability · See SoA static analysis · 213 story · 8, 65 Strategic IS Architecture · See SA Strategic Match · See SM Strength, Weakness, Opportunities, Threats · See SWOT Structured Query Language · See SQL Subject Matter Expert · See SME SVN · 115 SVS · 5, 45, 220 SWOT · 41, 42, 43, 61, 62, 69, 77, 146, 220 System Building Block · See SBB System Building Block Application · See SBB-A System Building Block Infrastructure · See SBB-I System Building Block Technology · See SBB-T System Integration Test · See SIT System of Engagement · See SoE System of Records · See SoR System Test · See ST Systems of Information · See SoI

T

Target Operating Model · See TOM task · 8, 65, 201, 207 TCO · 220 TCP · 220 TCP/IP · 54, 93 TDD · 99, 101, 116, 213, 220 Team Foundation Server · See TFS technical debt · 204, 205 technical debt backlog · 192 Technical Information Security Officer · See TISO Technical Uncertainty · See TU technology adaption curve · 213 technology executive · 213 template · 33 test - base · 85, 86 - case · 201, 202, 203 - harness · 213 - object-matrix · 103

- technique · 85 - technique matrix · 101, 102 Test Driven Development · See TDD tester · 205 TFS · 64, 220 The Agile Manifesto · 213 the ideal testing automation pyramid · 214 The Lean movement · 214 the non-ideal testing automation inverted pyramid \cdot 214 The Three Ways · 210, 214 theme \cdot 8, 67 theory of constraints · 214 Time To Market · See TTM TISO · 220 TMAP · 27, 66, 73 TOGAF model · 136 TOM · 43, 168, 178, 187, 220 tool-assisted code review · 214 Total Cost of Ownership · See TCO Toyota Kata · 215 Toyota Production System · See TPS TPS · 9, 46, 199, 220 traceability · 17, 97, 107, 121, 127, 128, 129, 157, 161, 190 transformation team · 215 Transmission Control Protocol · See TCP trunk · 212 T-shaped · 206 T-shaped pattern · 151 TSQL · 54, 93, 204 TTM · 220 TU · 220

U

UAT \cdot 102, 103, 220 UML \cdot 220 Unified Modeling Language \cdot See UML Unit Test \cdot See UT use case \cdot 32, 33, 35, 37, 38, 40, 77, 95, 105, 117, 125, 145, 155, 175 use case diagram \cdot 32, 35, 38 User Acceptance Test \cdot See UAT User eXperience Design \cdot See UX design UT \cdot 220 UX design \cdot 220

V

value chain · 2, 9, 44, 45, 46, 61, 69, 75, 77, 146, 165, 176

value stream · 2, 31, 41, 55, 57, 87, 205, 209, 212, 213, 215, 220 - canvas · 55, 69, 78, 83, 87 - manager · 213 - mapping · 69, 78, 146 - mapping model · 75 Value Stream Mapping · See VSM value system \cdot 2, 39 velocity · 202 version · 109 - control · 108, 110 - control system · 105, 110, 111, 112, 113, 114, 115, 116 - loss · 115 vertical splitting of feature · 215 virtualised environment · 215 vision · 16 visualisation · 215 Voice over Internet Protocol · See VOIP VOIP · 220 VSM · 161, 215, 220

W

walking skeleton · 215 WAN · 220 war room · 210 waste · 202, 204, 207, 208, 210, 214, 215 waste reduction · 215 Way of Working · See WoW Westrum · 210, 211 Wide Area Network · See WAN Windows Management Instrumentation • See WMI WIP · 220 WMI · 220 Work In Progress · See WIP work item · 8 workflow · 204 WoW · 15, 18, 19, 220

X

XML · 220 XML/SOAP · 54, 93 XP · 221

Ζ

Zachman · 7, 8

Epilogue

My experience is that the ideas I capture in an article or book continue to evolve. If you are going to work with a certain topic from this book in your own DevOps organisation, I advise you to contact me. Perhaps there are additional articles or experiences in this area that I can share with you. This also applies inversely proportionally. If you have any experiences that complement what is described in this book, I invite you to share them with me. You can reach me via my e-mail address bartb@dbmetrics.nl.

About the author



Drs. Ing. B. de Best RI has been working in ICT since 1985. He has mainly worked in the top 100 of Dutch business and government. He has held positions in all phases of system development, including operation and management, for 12 years. He then focused on the service management field. Currently, as a consultant, he fulfils all aspects of the knowledge lifecycle of service management, such as writing and providing training to ICT managers and service management design, improving management processes, outsourcing (parts of) the management organisation and reviewing and auditing management organisations. He graduated in management field at both HTS level and University level.

Other books by this author



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 bent 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 heraankopen, wordt hij gepromote 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 de 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 7150 1456



Cloud SLA

The best practices of cloud service level agreements

More and more organisations are opting to replace traditional ICT services with cloud services. Drawing up effective SLAs for traditional ICT services is a real challenge for many organisations. With the advent of cloud services, this initially seems much simpler, but soon the difficult questions such as data ownership, information links and security are addressed. This book describes what cloud services are. The risks that organisations run when entering into contracts and SLAs are discussed. Based on a long list of risks and countermeasures, this book also provides recommendations for the design and content of the various service level management documents for cloud services.

This book first defines the term 'cloud' and then describes various aspects such as cloud patterns and the role of a cloud broker. The core of the book concerns the discussion of contract aspects, service documents, service designs, risks, SLAs, and cloud governance. To enable the reader to immediately get started with cloud SLAs, the book also includes checklists of the following documents: Underpinning Contract (UC), Service Level Agreement (SLA), File Financial Agreements (DFA), Dossier Agreements and Procedures (DAP), External Spec Sheets (ESS) and Internal Spec Sheets (ISS).

Author	: Bart de Best
Publisher	: Leonon Media, 2014
ISBN (NL)	: 978 90 7150 1739
ISBN (UK)	: 978 94 9261 8009



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) Service Quality Plan (SQP)
- Document Financial Agreements (DFA)

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

- Service Catalogue ٠
- External Spec Sheet (ESS)
- Internal Spec Sheet (ISS)
- Service Improvement Program (SQP)



ICT Prestatie-indicatoren

De beheerorganisatie meetbaar gemaakt.

De laatste jaren is het maken van concrete afspraken over de ICTserviceverlening steeds belangrijker geworden. Belangrijke oorzaken hiervoor zijn onder meer de stringentere wet- en regelgeving, de hogere eisen die gesteld worden vanuit regievoering over uitbestede services en de toegenomen complexiteit van informatiesystemen. Om op de gewenste servicenormen te kunnen sturen, is het belangrijk om een Performance Measurement System (PMS) te ontwikkelen. Daarmee kunnen niet alleen de te leveren ICT-services worden gemeten, maar tevens de benodigde ICTorganisatie 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 7150 1470



Quality Control & Assurance

Kwaliteit op maat.

De business stelt steeds hogere eisen aan de ICT-services die ICTorganisaties 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 7150 1531



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 7150 1784



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 7150 1913



Agile Service Management with scrum

Towards a healthy balance between the dynamics of development and the stability of information management.

The application of Agile software development is booming. The terms Scrum and Kanban are already established in many organisations. Agile software development sets different requirements for the implementation of software management. Many organisations are therefore busy considering this new challenge. Especially the interaction between the Scrum development process and the management of the software that the Scrum development process has produced is an important aspect area. This book discusses precisely this interaction.

Examples of topics that are discussed are the service portfolio, SLAs and the handling of incidents and change requests. This book first defines the risk areas when introducing Scrum and Kanban. After that, the various Agile concepts and concepts are discussed. The implementation of Agile service management is described at both organisational and process level. The relevant risks have been identified for each management process. It is also indicated how this can be implemented within the context of scrum.

Author	: Bart de Best
Publisher	: Leonon Media, 2014 (NL), 2018 (UK)
ISBN (NL)	: 978 90 7150 1807
ISBN (UK)	: 978 94 9261 8085



Agile Service Management with Scrum in Practice

Towards a healthy balance between the dynamics of development and the stability of information management.

Many companies are in the process of applying Agile software development in the form of Scrum or Kanban or have already started using the new development process. Sooner or later, the question arises as to how this development process relates to the management processes. This interface has already been examined in the book 'Agile Service Management with scrum' and a number of risks per management process have been identified. Countermeasures that can be taken are also defined. These risks were presented in a survey of ten organisations, and they were asked how they dealt with these risks.

It was also investigated which Agile aspects are applied and in particular those of Scrum or Kanban. Finally, each organisation performed a maturity assessment for both the Agile development process and the change management process. This book is the report on the research into the collaboration of Agile software development and management processes in practice. The target audience of this book includes all parties involved in the application of Agile software development and who would like to know how colleagues have designed this crucial interface for successful service provision. This book also provides a brief description of each organisation about the way in which the Agile development process is designed.

Author	: Bart de Best
Publisher	: Leonon Media, 2015 (NL), 2018 (UK)
ISBN (NL)	: 978 90 7150 1845
ISBN (UK)	: 978 94 9261 8177



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 teams work together.

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

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



DevOps Architecture

DevOps Architecture Best Practices

The world of systems development is changing at a rapid pace. In addition, Development (Dev) and Operations (Ops) are increasingly integrated so that solutions can be offered to the customer faster and of better quality. The question is how within this new view of DevOps there room is for Agile architecture. This book answers this question by providing many examples of architectural principles and models that guide the organisation and operation of a DevOps organisation. Throughout the book, as much as possible per paragraph, an explanation is given based on an imaginary company Assuritas.

This book consists of several parts, which makes the book modular. So, it does not have to be read from A to Z. The brief outline of the case company is followed by a discussion of the DevOps organisation from an architectural perspective. Then the DevOps management facility is discussed. Both treatises are made transparent based on the case company. After discussing the integration of the Dev and Ops roles, there are two useful analysis tools to determine the maturity of DevOps. The book concludes with a case in which the choice for Agile documentation is made based on architectural principles and models. This work on DevOps architecture is an indispensable tool in the design and implementation of a DevOps service organisation.

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

Continuous Everything books



Continuous Architecture

A publication in the Continuous Everything series.

Continuous Architecture focuses on ensuring that the organizational strategy is achieved by providing direction for the innovation and management of the information provision required for this. A distinction is made between System of Records (chain applications), System of Engagement (single applications) and System of Services (service architecture for development and management). This book is a publication in the continuous everything series. The content consists of a discussion of the value streams for the realization of the SOR, SOE and SOS systems. It also contains example architecture principles and models for each Continuous Everything value stream.

This holistic approach to Continuous Architecture enables optimal and integrated implementation for both the development and management of the information provision of the entire organization that is necessary for the realization of the organizational strategy.

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



Continuous Planning

A publication in the Continuous Everything series.

Continuous Planning is an approach to get a grip on changes that are made in the information provision in order to realise the outcome improvement of the business processes and thus achieve the business goals. The approach is aimed at multiple levels, whereby an Agile planning technique is provided for each level that refines the higher-level planning. In this way, planning can be made at a strategic, tactical, and operational level and in an Agile manner that creates as little overhead as possible and as much value as possible. This book is a publication in the continuous everything series. The content consists of a discussion of planning techniques such as the balanced scorecard, enterprise architecture, product vision, roadmap, epic one pager, product backlog management, release

planning and sprint planning. It also indicates how these techniques are related to each other. In addition, this book indicates how to set up continuous planning in your organisation based on the change manager paradigm and architecture principles and models. With this integral Agile approach to planning, you have a powerful tool at your disposal to systematically approach your organisation's strategy and thereby realise your business goals.

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



Continuous SLA

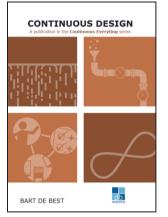
A publication in the Continuous Everything series.

Continuous SLA focuses on recognising risks that can harm the outcome of business processes (core value streams). These risks arise as a result of new construction and maintenance of information systems through Agile teams. Within the concept of Continuous SLA, these risks are analysed from different perspectives and provided with countermeasures by the DevOps team, also known as SLA controls. By making these SLA controls measurable, they become suitable planning objects that can be placed on the product backlog.

This book is a publication in the continuous everything series. The content consists of the discussion of techniques to identify and

manage risks such as the use of Lean indicators, value stream mapping and information, application and technical architecture building blocks. In addition to the core value streams, the enable value streams such as management, information security and development value streams are also examined for risks that directly or indirectly harm the outcome. The recognised SLA controls are anchored in the Agile way of working by deepening the collaboration between, among others, the product owner and service level manager. This integrated approach to SLA controls makes it possible to get a grip on quality in Agile projects.

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



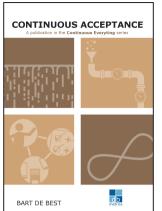
Continuous Design

A publication in the Continuous Everything series.

Continuous Design is an approach that aims to allow DevOps teams to briefly think in advance about the contours of the information system to be realised and to allow the design to grow during the Agile project (emerging design). This prevents interface risks and guarantees essential knowledge transfer to support management and compliance with legislation and regulations. Elements that guarantee the continuity of an organisation. This book is a publication in the continuous everything series. The content consists of the continuous design pyramid model in which the following design views are defined: business, solution, design, requirements, test, and code view.

The continuous design encompasses the entire lifecycle of the information system. The first three views are completed based on modern design techniques such as value stream mapping and use cases. However, the emphasis of the effective application of a continuous design lies in the realisation of the information system, namely by integrating the design in the Behavior Driven Development and Test-Driven Development as well as in continuous documentation. With this Agile approach to design you have a powerful tool at your disposal to get a grip on an Agile development project.

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



Continuous Acceptance

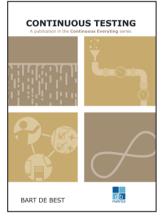
A publication in the Continuous Everything series.

Continuous Acceptance focuses on accepting new and modified products and services produced in an Agile environment.

In this CE value stream, the specific acceptance criteria are derived from the business value streams by looking for the risks that the business goals are not achieved. The countermeasures for these risks are tested for effectiveness through acceptance tests. By analogy, the generic acceptance criteria are derived from the CE value streams that flesh out the DevOps Lemniscate. This book is a publication in the continuous everything series. The content consists of the discussion of the derivation of acceptance criteria.

An example elaboration is also given for specific acceptance criteria and a number of generic acceptance criteria are given for the following value streams: Continuous Planning, Continuous Design, Continuous Testing, Continuous Integration, Continuous AI, Continuous Deployment, Continuous Monitoring, Continuous Learning, Continuous Security, Continuous Auditing, Continuous SLA and Continuous Assessment. This gives you a tool to get a grip on the acceptance of applications and services.

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



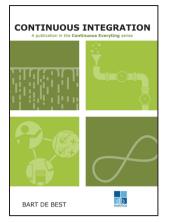
Continuous Testing

A publication in the Continuous Everything series.

Continuous Testing is an approach that aims to provide rapid feedback in the software development process by defining the 'what' and 'how' questions as test cases before starting to build the solution. As a result, the concepts of requirements, test cases and acceptance criteria are integrated in one approach. The term 'continuous' refers to the application of test management in all phases of the deployment pipeline, from requirements to production. The term 'continuous' also includes the aspects People, Process and Technology. This makes test management holistic. This book is a publication in the continuous everything series. The content consists of treating continuous testing based on a definition, business case, architecture, design, and best practices.

Concepts discussed are: the change paradigm, the ideal test pyramid, test metadata, Behavior Driven Development (BDD), Test Driven Development (TDD), test policies, test techniques, test tools and the role of unit test cases in continuous testing. In this way you are quickly up to date in the field of DevOps developments and in the field of continuous testing.

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



Continuous Integration

A publication in the Continuous Everything series.

Continuous Integration is a holistic Lean software development approach that aims to produce and put into production continuous software in an incremental and iterative way, where waste reduction is of paramount importance.

The word 'holistic' refers to the PPT concepts: People (multiple expert), Process (knowledge of business and management processes) and Technology (application and infrastructure programming). The incremental and iterative method makes fast feedback possible because functionalities can be put into production earlier. This reduces waste because defects are found earlier and can be repaired faster.

This book is a publication in the continuous everything series. The content consists of treating continuous integration based on a definition, business case, architecture, design, and best practices. Concepts discussed here are the change paradigm, the application of continuous integration, use of repositories, code quality, green code, green build, refactoring security-based development and built-in failure mode. In this way you are quickly up to date in the field of DevOps developments with regard to continuous integration.

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



Continuous AI

A publication in the Continuous Everything series.

Continuous AI focuses on increasing the outcome of CE aspect areas such as the value streams Continuous Testing and Continuous Integration. The bottlenecks of these value streams are located in the form of limitations (performance) and boundaries (functionality). These bottlenecks can be reduced or removed through AI application areas, such as using Machine Learning (ML) and Natural Language Processing (NLP). This book is a publication in the continuous everything series.

The content consists of the discussion of AI application areas for all published CE aspect areas. For each step of each value stream, it is indicated what the possibilities of AI are now or in the future.

The value streams involved are: Continuous Planning, Continuous Design, Continuous Testing, Continuous Integration, Continuous Deployment, Continuous Monitoring, Continuous Learning, Continuous Security, Continuous Auditing, Continuous SLA and Continuous Assessment. This gives you a tool to apply AI in your organisation in a structured and effective way.

Author	: Bart de Best
Publisher	: Leonon Media, 2022
ISBN (NL)	: 978 94 91480 294
ISBN (UK)	: 978 94 91480 300



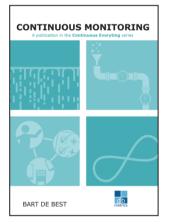
Continuous Deployment

A publication in the Continuous Everything series.

Continuous Deployment is a holistic Lean production approach that aims to deploy and release continuous software in an incremental and iterative way, where time to market and high quality are of paramount importance. The word 'holistic' refers to the PPT concepts: People (multiple expert), Process (knowledge of business and management processes) and Technology (application and infrastructure programming. The incremental and iterative deployments enable fast feedback because errors are more likely to are observed in production of the CI/CD secure pipeline, making recovery actions faster and cheaper, leading to a waste reduction.

This book is a publication in the continuous everything series. The content consists of treating continuous deployment based on a definition, business case, architecture, design, and best practices. Concepts that are discussed here are the change paradigm, the application of continuous deployment, a step-by-step plan for the systematic arrangement of continuous deployment and many patterns to allow deployments to take place. In this way you are quickly up to date in the field of DevOps developments in the field of continuous deployment.

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



Continuous Monitoring

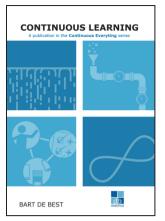
A publication in the Continuous Everything series.

Continuous Monitoring is an approach to get a grip on both core value streams (business processes) and enable value streams that support these core value streams. Continuous Monitoring differs from classical monitoring by its focus on outcome improvement and the holistic scope with which value streams are measured, i.e. the entire CI/CD secure pipeline for all three perspectives of PPT: People, Process and Technology.

The approach includes People, Process and Technology, which makes it possible to identify and eliminate or mitigate the bottlenecks in your value streams.

This book is a publication in the continuous everything series. The content consists of a discussion of the monitor functions defined in the continuous monitoring layer model. This layer model classifies the monitoring tools available on the market. Each monitor archetype is defined in this book in terms of definition, objective, measurement attributes, requirements, examples, and best practices. This book also indicates how to set up continuous monitoring in your organisation based on the change manager paradigm and architecture principles and models. With this integral agile approach to monitoring you have a powerful tool at your disposal to set up the controls for the control of your value streams.

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



Continuous Learning

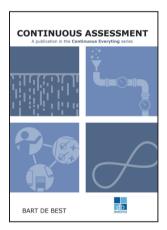
A publication in the Continuous Everything series.

Continuous Learning is an approach to get a grip on the competences needed to realise your organisation's strategy. To this end, continuous learning offers Human Resource Management an approach that explores the organisational needs and competences step by step and converts these needs into competency profiles.

A competency profile is defined here as the set of knowledge, skills and behavior at a certain Bloom level that produces a certain result. Competency profiles are then merged into roles that in turn form functions. In this way an Agile job house is obtained. This book is a publication in the continuous everything series.

The content consists of a discussion of the continuous learning model that helps you to translate a value chain strategy step by step into a personal roadmap for employees. This book also indicates how to organise Continuous Learning in your organisation based on the paradigm of the change manager and architecture principles and models. With this agile approach to HRM you have a powerful tool to get the competences to the desired level of your organisation.

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



Continuous Assessment

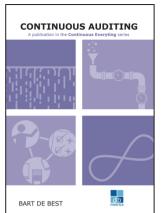
A publication in the Continuous Everything series.

Continuous Assessment is an approach that aims to allow DevOps teams to continuously develop in terms of knowledge and skills in the field of business, development, operations, and security. This book provides a tool to make the DevOps teams aware where they stand in terms of development and what next steps they can take to develop. This book is a publication in the continuous everything series.

The content consists of the business case for continuous assessment, the architecture of the two assessment models and the assessment questionnaires.

The DevOps Cube model is based on the idea that DevOps can be viewed from six different perspectives of a cube, namely: 'Flow', 'Feedback', 'continuous learning', 'Governance', 'Pipeline' and 'QA'. The DevOps CE model is based on the continuous everything perspectives, namely: 'continuous integration', 'continuous deployment', 'continuous testing', 'continuous monitoring', continuous documentation' and 'continuous learning'. This book is an excellent mirror for any DevOps team that wants to quickly form a complete picture of DevOps best practices to be adopted.

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



Continuous Auditing

A publication in the Continuous Everything series.

Continuous Auditing is an approach that aims to enable DevOps teams to demonstrate in a short cyclical way that they are in control when realising, putting into production, and managing the new or modified products and services at a rapid pace.

As a result, compliance risks are prevented by already thinking about which risks to mitigate or eliminate from the requirements and the design based on them.

This book is a publication in the continuous everything series. The content consists of an explanation of the continuous auditing pyramid model that describes the six steps to give substance to

continuous auditing, namely: determining scope, determining goals, identifying risks, realising controls, setting up monitoring facilities and demonstrating effectiveness of controls.

The Continuous Auditing concept thus encompasses the entire lifecycle of risk management. As a result, the risks are continuously under control. With this Agile approach of auditing, you have a powerful tool to get a grip on the compliancy of your Agile system development and management.

Author	: Bart de Best
Publisher	: Leonon Media, 2022
ISBN (NL)	: 978 94 92618 542
ISBN (UK)	: 978 94 92618 757



Continuous Security

A publication in the Continuous Everything series.

Continuous Security is an approach that aims to keep an organisation in control from three perspectives:

- The business perspective: Business value streams are in control of the identified risks by continuously testing the effectiveness of the controls deployed and recording evidence.
- The development perspective: Development value streams are in control by integrally including the non-functional requirements for information security in the development.
- The operations perspective: Operations value streams are in control for the production of the new and changed ICT services through an adequate design of the CI/CD secure pipeline in which controls automatically test the non-functional require-

ments. This book is a publication in the continuous everything series. The content consists of a discussion of the application of ISO 27001 on the basis of three sets of security practices, namely Governance, Risk and Quality. The practices are provided with a definition and objective. In addition, examples and best practices are given.

The continuous security concept is designed to be used in Agile Scrum (development) and DevOps (Development & Operations) environments. To this end, it connects seamlessly to common Agile management models. This Agile approach to information security provides you with a powerful tool to get a grip on the compliance of your Agile system development and management.

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



Continuous Development

A publication in the Continuous Everything series.

Continuous Everything is the collective name for all Continuous Developments that are currently going on in the DevOps world. By placing these under one heading, structure can be applied to individual developments and best practices can be defined on the basis of patterns.

The term 'Continuous' includes the terms: outcome driven development, incremental & iterative working, waste reduction through a Lean approach, holistic working by including people, process, partner & technology in the scope and giving continuous attention to a deliverable (product or service) across the entire lifecycle from an end-to-end approach.

This book is a collection of four Continuous Everything books, namely: Continuous Planning, Continuous Design, Continuous Testing and Continuous Integration. For each Continuous Everything aspect area it is indicated how to organise it in your organisation based on the change manager paradigm and architecture principles and models. The best practices are also discussed per aspect area. With this book in hand, you have a powerful tool to further your DevOps skills.

Author	: Bart de Best
Publisher	: Leonon Media, 2022
ISBN (NL)	: 978 94 92618 641
ISBN (UK)	: 978 94 92618 764
	. 570 54 52010 704



Continuous Operations

A publication in the Continuous Everything series.

Continuous Everything is the collective name for all Continuous Developments that are currently going on in the DevOps world. By placing these under one heading, structure can be applied to individual developments and best practices can be defined on the basis of patterns.

The term 'Continuous' includes the terms: outcome driven development, incremental & iterative working, waste reduction through a Lean approach, holistic working by including people, process, partner & technology in the scope and giving continuous attention to a deliverable (product or service) across the entire lifecycle from an end-to-end approach.

This book is a collection of four Continuous Everything books, namely: Continuous Deployment, Continuous Monitoring, Continuous Learning and Continuous Assessment. For each Continuous Everything aspect area it is indicated how to organise it in your organisation based on the change manager paradigm and architecture principles and models. The best practices are also discussed per aspect area. With this book in hand, you have a powerful tool to further your DevOps skills.

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



Continuous Control

A publication in the Continuous Everything series.

Continuous Everything is the collective name for all Continuous Developments that are currently going on in the DevOps world. By placing these under one heading, structure can be applied to individual developments and best practices can be defined on the basis of patterns.

The term 'Continuous' includes the terms: outcome driven development, incremental & iterative working, waste reduction through a Lean approach, holistic working by including people, process, partner & technology in the scope and giving continuous attention to a deliverable product or service across the entire lifecycle from an end-to-end approach.

This book is a collection of three Continuous Everything books, namely: Continuous Assessment, Continuous Security, Continuous Audit. For each Continuous Everything aspect area it is indicated how to organise it in your organisation based on the change manager paradigm and architecture principles and models. The best practices are also discussed per aspect area. With this book in hand, you have a powerful tool to further your DevOps skills.

Author Publisher ISBN (NL) ISBN (UK)



: Bart de Best : Leonon Media, 2022 : 978 94 91480 195 : 978 94 91480 201

Continuous Everything

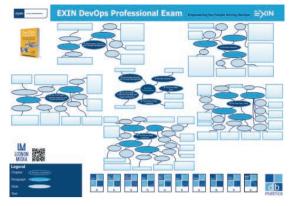
A publication in the Continuous Everything series.

Continuous Everything is the collective name for all Continuous Developments that are currently going on in the DevOps world. By placing these under one heading, structure can be applied to individual developments and best practices can be defined on the basis of patterns.

The term 'Continuous' includes the terms: outcome driven development, incremental & iterative working, waste reduction through a Lean approach, holistic working by including people, process, partner & technology in the scope and giving continuous attention to a deliverable product or service across the entire lifecycle from an end-to-end approach.

This book is a collection of eight Continuous Everything books, namely: Continuous Planning, Continuous Design, Continuous Testing, Continuous Integration, Continuous Deployment, Continuous Monitoring, Continuous Learning and Continuous Assessment. For each Continuous Everything aspect area it is indicated how to organise it in your organisation based on the change manager paradigm and architecture principles and models. The best practices are also discussed per aspect area. With this book in hand, you have a powerful tool to further your DevOps skills.

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



Author Publisher Ordering : Bart de Best

: Leonon Media, 2018

: info@leonon.nl

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.

CONTINUOUS ARCHITECTURE

A publication in the Continuous Everyting series

Bart de Best

Continuous Architecture focuses on ensuring that the organizational strategy is achieved by providing direction for the innovation and management of the information provision required for this.

A distinction is made between System of Records (chain applications), System of Engagement (single applications) and System of Services (service architecture for development and management)..

This book is a publication in the continuous everything series. The content consists of a discussion of the value streams for the realization of the SOR, SOE and SOS systems. It also contains example architecture principles and models for each Continuous Everything value stream. This holistic approach to Continuous Architecture enables optimal and integrated implementation for both the development and management of the information provision of the entire organization that is necessary for the realization of the organizational strategy.



