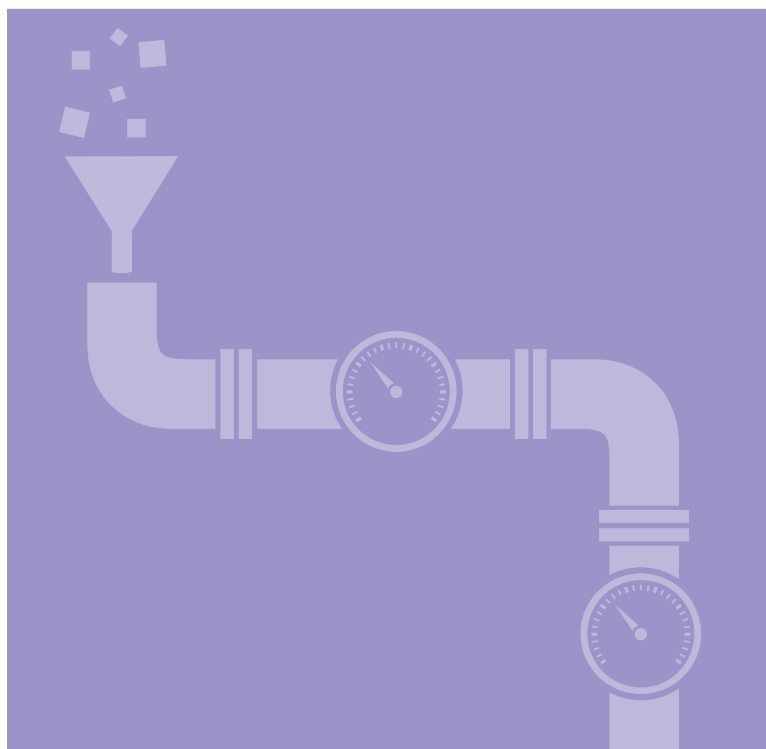


CONTINUOUS CONTROL

The complete ABC of DevOps Control



BART DE BEST



DevOps Continuous Control

The complete ABC of DevOps Control

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 : info@leonon.nl
Sales questions : verkoop@leonon.nl
Manuscript / Author : redactie@leonon.nl

© 2023 Leonon Media

Cover design : Eric Coenders, IanusWeb, Nijmegen
Production : Printforce B.V., Culemborg

Title : DevOps Continuous Control
Subtitle : The complete ABC of DevOps Control
Date : 28 May 2023
Author : Bart de Best
Publisher : Leonon Media
ISBN13 : 978 94 91480 201
Edition : First press, eighth edition, 28 May 2023

© 2023, 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	INTRODUCTION.....	1
1.1	OBJECTIVE.....	1
1.2	TARGET GROUP.....	1
1.3	BACKGROUND.....	1
1.4	STRUCTURE.....	2
1.5	APPENDICES.....	3
1.6	READING GUIDELINES.....	4

PART I. CONTINUOUS AUDITING

1	PREFACE CONTINUOUS AUDITING.....	7
1.1	OBJECTIVE.....	7
1.2	POSITIONING.....	7
1.3	STRUCTURE.....	9
2	BASIC CONCEPTS AND BASIC TERMS.....	11
2.1	BASIC CONCEPTS.....	11
2.2	BASIC TERMS.....	12
3	CONTINUOUS AUDITING DEFINITION.....	17
3.1	BACKGROUND.....	17
3.2	DEFINITION.....	17
3.3	APPLICATION.....	17
4	CONTINUOUS AUDITING ANCHORING.....	21
4.1	THE CHANGE PARADIGM.....	21
4.2	VISION.....	22
4.3	POWER.....	23
4.4	ORGANISATION.....	27
4.5	RESOURCES.....	28
5	CONTINUOUS AUDITING ARCHITECTURE.....	31
5.1	ARCHITECTURE PRINCIPLES.....	31
5.2	ARCHITECTURE MODELS.....	34
6	CONTINUOUS AUDITING DESIGN.....	39
6.1	CONTINUOUS AUDITING VALUE STREAM.....	39
6.2	CONTINUOUS AUDITING USE CASE DIAGRAM.....	39
6.3	CONTINUOUS AUDITING USE CASE.....	40
7	CONTINUOUS AUDITING BEST PRACTICES.....	45
7.1	DETERMINE SCOPE TOM.....	45
7.2	SELECT TARGETS.....	49
7.3	IDENTIFY RISKS.....	54
7.4	REALISE CONTROLS.....	59
7.5	MONITOR CONTROLS.....	62
7.6	DEMONSTRATE EFFECTIVENESS CONTROLS.....	64
8	ARTICLE 1. CA CONCEPT.....	67

8.1	INTRODUCTION	67
8.2	DEFINITION OF THE PROBLEM	67
8.3	SOLUTIONS	67
8.4	CONTINUOUS AUDITING, THE SOLUTION	69
8.5	CONCLUSION	71
8.6	WHY ONE AGILE WAY OF WORKING?	72
9	ARTICLE 2. CA TOOL DESIGNED	75
9.1	SUMMARY	75
9.2	INTRODUCTION	75
9.3	PROBLEM STATEMENT	76
9.4	CA-TOOL QUESTIONS	77
9.5	CA-TOOL REQUIREMENTS	78
9.6	HOW TO DESIGN THE REQUIREMENTS?	82
9.7	EXAMPLES OF A CA TOOL DESIGN	82
9.8	CA-TOOL APPROACH	83
9.9	EXAMPLES OF CA-TOOLS ON THE MARKET	84
10	ARTIKEL 3. CA IMPLEMENTED	85
10.1	INTRODUCTION	85
10.2	MVP INTRODUCTION	85
10.3	MVP EXPLAINED	86
10.4	FINDINGS AND CONCLUSION	88

PART II. CONTINUOUS SECURITY

1	INTRODUCTION CONTINUOUS SECURITY	93
1.1	GOAL	93
1.2	POSITIONING	93
1.3	STRUCTURE	95
2	BASIC CONCEPTS AND BASIC TERMS.....	97
2.1	BASIC CONCEPTS	97
2.2	BASIC TERMS	103
3	CONTINUOUS SECURITY DEFINITION	107
3.1	BACKGROUND	107
3.2	DEFINITION	107
3.3	APPLICATION	107
4	CONTINUOUS SECURITY ANCHORAGE	111
4.1	THE CHANGE PARADIGM	111
4.2	VISION	112
4.3	POWER.....	113
4.4	ORGANISATION	117
4.5	RESOURCES	118
5	CONTINUOUS SECURITY ARCHITECTURE.....	121
5.1	ARCHITECTURE PRINCIPLES	121
5.2	ARCHITECTURE MODELS	124
6	CONTINUOUS SECURITY DESIGN.....	135
6.1	CONTINUOUS SECURITY VALUE STREAM	135

6.2	CONTINUOUS SECURITY USE CASE DIAGRAM	136
6.3	CONTINUOUS SECURITY USE CASE	137
7	CONTINUOUS SECURITY BEST PRACTICES	143
7.1	BEST PRACTICES	143
7.2	VALUE STREAM EXAMPLES	143
8	GOVERNANCE SECURITY PRACTICES	145
8.1	SCOPE GOVERNANCE SECURITY PRACTICES	145
8.2	GET TOP MANAGEMENT COMMITMENT	145
8.3	DETERMINE INTERESTED PARTIES	148
8.4	DETERMINE SCOPE	151
8.5	DETERMINE GOALS	153
8.6	DETERMINE INFORMATION SECURITY POLICIES	155
9	RISK SECURITY PRACTICES	159
9.1	SCOPE RISK SECURITY PRACTICES	159
9.2	DETERMINE ISSUES - INTERNAL	160
9.3	DETERMINE ISSUES - EXTERNAL	164
9.4	DETERMINE CRAMM ISSUES	168
9.5	DETERMINE RISK CRITERIA	172
9.6	DETERMINE INFORMATION ASSETS	181
9.7	IDENTIFY RISKS	183
9.8	PERFORM RISK ASSESSMENT	186
9.9	PERFORM RISK TREATMENT - OPTIONS	189
9.10	DETERMINE RISK TREATMENT - CONTROLS	191
9.11	PERFORM RISK TREATMENT – EXISTING CONTROLS	194
9.12	REALISE CONTROLS	195
10	QUALITY SECURITY PRACTICES	199
10.1	SCOPE QUALITY SECURITY PRACTICES	199
10.2	MONITOR EFFECTIVENESS CONTROLS	199
10.3	PERFORM INTERNAL AUDIT - PLAN	203
10.4	PERFORM INTERNAL AUDIT - CRITERIA	205
10.5	PERFORM INTERNAL AUDIT - PERFORMANCE	206
10.6	PERFORM INTERNAL AUDIT - REPORT	208
10.7	IMPROVE CONTINUOUS - INCIDENTS	209
10.8	IMPROVE CONTINUOUS – NON CONFORMITIES	211
10.9	IMPROVE CONTINUOUS – CSI	212
11	CONTINUOUS SECURITY VERSUS AGILE SCRUM	215
11.1	POSITIONING	215
11.2	AGILE MANIFESTO	215
11.3	AGILE METHODS	218
11.4	AGILE SCRUM	218
11.5	CONTINUOUS SECURITY IN AGILE SCRUM	222
11.6	THE DIFFERENCE	226
12	CONTINUOUS SECURITY VERSUS DEVOPS	227
12.1	DEVOPS POSITIONING	227
12.2	DEVOPS CONCEPT	227
12.3	CONTINUOUS SECURITY IN DEVOPS	230
12.4	THE DIFFERENCE	232

PART III. CONTINUOUS SLA

1	INTRODUCTION CONTINUOUS SLA	237
1.1	GOAL	237
1.2	BACKGROUND	237
1.3	STRUCTURE	238
1.4	READING GUIDELINES	238
2	BASIC CONCEPTS AND BASIC TERMS.....	239
2.1	BASIC CONCEPTS	239
2.2	BASIC TERMS	244
3	CONTINUOUS SLA DEFINITION	251
3.1	BACKGROUND	251
3.2	DEFINITION	251
3.3	APPLICATION	251
4	CONTINUOUS SLA ANCHORING	255
4.1	THE CHANGE PARADIGM	255
4.2	VISION	256
4.3	POWER.....	257
4.4	ORGANISATION	259
4.5	RESOURCES	261
5	CONTINUOUS SLA ARCHITECTURE	263
5.1	ARCHITECTURE PRINCIPLES	263
5.2	ARCHITECTURE MODELS	265
6	CONTINUOUS SLA DESIGN	271
6.1	CONTINUOUS SLA VALUE STREAM	271
6.2	CONTINUOUS SLA USE CASE DIAGRAM	271
6.3	CONTINUOUS SLA USE CASE.....	272
7	CONTINUOUS SLA MODEL	277
7.1	THE CONTINUOUS SLA MODEL	277
7.2	RISK SOURCES.....	277
7.3	THE CONTINUOUS SLA RASCI MODEL.....	278
7.4	JOB DESCRIPTION SERVICE LEVEL MANAGER.....	280
8	DETERMINE VALUE STREAM SCOPE (1).....	281
8.1	VALUE CHAIN.....	281
9	DETERMINE VALUE STREAM GOALS (2)	283
9.1	SWOT ANALYSIS.....	283
9.2	VALUE STREAM GOVERNANCE MODEL	283
10	DETERMINE VALUE STREAM MAPPING (3).....	285
10.1	ROADMAP TO VALUE.....	285
10.2	VALUE STREAM MAPPING	285
11	DETERMINE VALUE STREAM CONTROLS (4)	287
11.1	ROADMAP TO VALUE.....	287
11.2	VALUE STREAM CANVAS.....	287
11.3	ARCHITECTURE BUILDING BLOCKS	290
11.4	EXAMPLES OF RISKS AND SLA CONTROLS	293

12	AGREE SLA NORMS (5)	295
12.1	SERVICE NORMS	295
13	MONITOR SLA CONTROLS (6)	297
13.1	MONITOR CLASSIFICATION MODEL	297
13.2	MONITOR CLASSIFICATION MODEL	298
14	COFFEE CASE	299
14.1	DETERMINE VALUE STREAM SCOPE (1)	299
14.2	DETERMINE VALUE STREAM GOALS (2)	301
14.3	DETERMINE VALUE STREAM MAPPING (3)	304
14.4	DETERMINE SLA CONTROLS (4)	305
14.5	AGREE SLA NORMS (5)	308
14.6	MONITOR SLA CONTROLS (6)	309

PART IV. CONTINUOUS ASSESSMENT

1	INTRODUCTION CONTINUOUS ASSESSMENT	313
1.1	GOAL	313
1.2	POSITIONING	313
1.3	STRUCTURE	313
2	BASIC CONCEPTS AND BASIC TERMS	315
2.1	BASIC CONCEPTS	315
2.2	BASIC TERMS	316
3	CONTINUOUS ASSESSMENT DEFINITION	319
3.1	BACKGROUND	319
3.2	DEFINITION	319
3.3	APPLICATION	319
4	CONTINUOUS ASSESSMENT ANCHORING	321
4.1	THE CHANGE PARADIGM	321
4.2	VISION	322
4.3	POWER	323
4.4	ORGANISATION	326
4.5	RESOURCES	327
5	CONTINUOUS ASSESSMENT ARCHITECTURE	329
5.1	ARCHITECTURE PRINCIPLES	329
5.2	ARCHITECTURE MODELS	331
6	CONTINUOUS ASSESSMENT DESIGN	337
6.1	CONTINUOUS ASSESSMENT VALUE STREAM	337
6.2	CONTINUOUS ASSESSMENT USE CASE DIAGRAM	337
6.3	CONTINUOUS ASSESSMENT USE CASE	338
7	DEVOPS CUBE ASSESSMENT MODEL	343
7.1	SIDE 1 – FLOW	343
7.2	SIDE 2 – FEEDBACK	344
7.3	SIDE 3 – CONTINUAL LEARNING AND EXPERIMENTING	346
7.4	SIDE 4 – GOVERNANCE	346
7.5	SIDE 5 – E2E DEPLOYMENT PIPELINE	347

7.6	SIDE 6 – QUALITY ASSURANCE.....	348
8	DEVOPS CE ASSESSMENT MODEL	351
8.1	DEVOPS CE MODEL, CP	351
8.2	DEVOPS CE MODEL, CN.....	353
8.3	DEVOPS CE MODEL, CT	356
8.4	DEVOPS CE MODEL, CI	359
8.5	DEVOPS CE MODEL, CD.....	362
8.6	DEVOPS CE MODEL, CM.....	365
8.7	DEVOPS CE MODEL, CL	368
8.8	DEVOPS CE MODEL, CY	371
8.9	OVERVIEW BY ASPECT AREA.....	374
8.10	ADDITIONAL ASSESSMENTS	379
	APPENDIX A, LITERATURE LIST	389
	APPENDIX B, GLOSSARY.....	393
	APPENDIX C, ABBREVIATIONS.....	409
	APPENDIX D, WEBSITES	413
	APPENDIX E, INDEX.....	415

Figures

FIGURE 1-1, DEVOPS LEMNISCATE.	1
-------------------------------------	---

PART I. CONTINUOUS AUDITING

FIGURE 1-1, SoR, SOE EN SOI (SOURCE HSO THE RESULT COMPANY).	8
FIGURE 2-1, CONTINUOUS AUDITING PYRAMID.	11
FIGURE 2-2, CONTINUOUS CONTROL.	12
FIGURE 2-3, VALUE CHAIN OF PORTER, SOURCE: [BOOK MICHAEL PORTER].	13
FIGURE 2-4, RECURSIVE VALUE CHAIN OF PORTER, SOURCE: [BOOK MICHAEL PORTER].	14
FIGURE 2-5, RECURSIVE VALUE CHAIN OF PORTER, SOURCE: [BOOK MICHAEL PORTER].	15
FIGURE 2-6, THE CONSTRUCTION OF A VALUE SYSTEM.	15
FIGURE 2-7, THE CONSTRUCTION OF A BUSINESS VALUE CHAIN.	16
FIGURE 4-1, THE CHANGE PARADIGM.	21
FIGURE 4-2, CHANGE PARADIGM - VISION.	22
FIGURE 4-3, CHANGE PARADIGM - POWER.	24
FIGURE 4-4, CHANGE PARADIGM - ORGANISATION.	27
FIGURE 4-5, CHANGE PARADIGM - RESOURCES.	28
FIGURE 5-1, CONTINUOUS AUDITING PYRAMID.	35
FIGURE 5-2, CONTINUOUS AUDITING PYRAMID DEPICTED ON THE DEVOPS LEMNISCATE.	35
FIGURE 5-3, CONTINUOUS AUDITING PYRAMID WITH DELIVERABLES AND QUESTIONS TO BE ANSWERED.	36
FIGURE 5-4, CONTINUOUS AUDITING PYRAMID MODEL DEPICTED ON CONTINUOUS CONTROL MODEL.	37
FIGURE 5-5, QUALITY CONTROL & ASSURANCE MODEL.	37
FIGURE 6-1, CONTINUOUS AUDITING VALUE STREAM.	39
FIGURE 6-2, USE CASE DIAGRAM FOR CONTINUOUS AUDITING.	40
FIGURE 7-1, CONTINUOUS AUDITING VALUE STREAM.	45
FIGURE 7-2, BUSINESS MODEL CANVAS.	47
FIGURE 7-3, SYSTEM CONTEXT DIAGRAM TEMPLATE.	48
FIGURE 7-4, BALANCED SCORECARD [KAPLAN 2004].	49
FIGURE 7-5, CSF-SCHEME.	51
FIGURE 7-6, CASCADATION OF THE BALANCED-SCORECARD.	51
FIGURE 7-7, VALUE STREAM CANVAS.	52
FIGURE 7-8, RISK LIFECYCLE MANAGEMENT.	55
FIGURE 7-9, ROADMAP TO VALUE [LAYTON 2017].	61
FIGURE 7-10, MONITOR LAYER MODEL.	64
FIGURE 7-11, THE CONCEPT OF CONTINUOUS AUDITING.	66
FIGURE 8-1, VALUE SYSTEMS.	68
FIGURE 8-2, THE CONCEPT OF CONTINUOUS AUDITING.	70
FIGURE 8-3, THE VALUE SYSTEMS VIEWS.	73
FIGURE 9-1, CONTINUOUS AUDITING CONCEPT.	76
FIGURE 9-2, THE VALUE STREAMS OF THE CA VALUE SYSTEM.	78
FIGURE 9-3, THE USE CASE DIAGRAMS FOR EVIDENCE MANAGEMENT (VS-02 VALUE STREAM).	79
FIGURE 9-4, CA-TOOL DATA MODEL.	83
FIGURE 10-1, USER INTERFACE CA-TOOL.	86
FIGURE 10-2, CA-TOOL ENTITY RELATIONSHIP DIAGRAM.	88

PART II. CONTINUOUS SECURITY

FIGURE 1-1, SoR, SOE EN SOI (SOURCE HSO THE RESULT COMPANY).	94
---	----

FIGURE 2-1, CONTINUOUS CONTROL.97

FIGURE 2-2, CONTINUOUS SECURITY PYRAMID.98

FIGURE 2-3, VALUE CHAIN OF PORTER, BRON: [BOEK MICHAEL PORTER].99

FIGURE 2-4, RECURSIVE VALUE CHAIN OF PORTER, BRON: [MICHAEL PORTER 1998].100

FIGURE 2-5, RECURSIVE VALUE CHAIN OF PORTER, BRON: [MICHAEL PORTER].101

FIGURE 2-6, THE COMPOSITION A VALUE SYSTEM.101

FIGURE 2-7, THE STRUCTURE OF A BUSINESS VALUE CHAIN.102

FIGURE 2-8, THE THREE PERSPECTIVES OF INFORMATION SECURITY.103

FIGURE 2-9, RISK TERMS.104

FIGURE 2-10, VALUE SYSTEM TERMS.105

FIGURE 4-1, CHANGE PARADIGM.111

FIGURE 4-2, THE CHANGE PARADIGM - VISION.112

FIGURE 4-3, THE CHANGE PARADIGM - POWER.114

FIGURE 4-4, THE CHANGE PARADIGM - ORGANISATION.117

FIGURE 4-5, THE CHANGE PARADIGM - RESOURCES.118

FIGURE 5-1, CONTINUOUS SECURITY PYRAMID.125

FIGURE 5-2, CONTINUOUS SECURITY PYRAMID DEPICTED ON THE DEVOPS LEMNISCATE.125

FIGURE 5-3, CONTINUOUS SECURITY PYRAMID WITH DELIVERABLES AND QUESTIONS TO BE ANSWERED.126

FIGURE 5-4, CONTINUOUS SECURITY PYRAMID MODEL MAPPED TO CONTINUOUS CONTROL MODEL.127

FIGURE 5-5, QUALITY CONTROL & ASSURANCE MODEL.127

FIGURE 5-6, RECURSIVE VALUE CHAIN.128

FIGURE 5-7, INFORMATION SECURITY VALUE CHAIN.129

FIGURE 5-8, INFORMATION SECURITY VALUE SYSTEM.129

FIGURE 5-9, INFORMATION SECURITY PRACTICES.130

FIGURE 5-10, INFORMATION SECURITY VALUE SYSTEM OVERVIEW.131

FIGURE 5-11, SERVICE VALUE CHAIN.131

FIGURE 5-12, DEVELOPMENT VALUE CHAIN.132

FIGURE 5-13, CONTINUOUS SECURITY PYRAMID DEPICTED ON THE ISVS, DVS AND SVS MODELS.132

FIGURE 5-14, INFORMATION SECURITY PERSPECTIVES.133

FIGURE 6-1, CONTINUOUS SECURITY VALUE STREAM.135

FIGURE 6-2, USE CASE DIAGRAM FOR CONTINUOUS SECURITY.137

FIGURE 7-1, INFORMATION SECURITY PRACTICES.143

FIGURE 7-2, INFORMATION SECURITY VALUE STREAMS.144

FIGURE 8-1, GOVERNANCE SECURITY PRACTICES.145

FIGURE 9-1, RISK SECURITY PRACTICES.159

FIGURE 9-2, CRAMM MODEL.171

FIGURE 9-3, ASSET REGISTER.183

FIGURE 9-4, RISK LIFECYCLE.185

FIGURE 10-1, QUALITY SECURITY PRACTICES.199

FIGURE 10-2, MONITOR ARCHITECTURE FOR MONITORING THE EFFECTIVENESS OF CONTROLS.202

FIGURE 11-1, POSITIONING AGILE AND AGILE SCRUM.215

FIGURE 11-2, AGILE SCRUM DEVELOPMENT PROCESS.219

FIGURE 11-3, AGILE SCRUM TEAMS.220

FIGURE 11-4, CONTINUOUS SECURITY USE CASE DIAGRAM.224

FIGURE 12-1, POSITIONING DEVOPS.227

FIGURE 12-2, CONTINUOUS SECURITY DEPICTED ON THE DEVOPS LEMNISCATE.228

FIGURE 12-3, CONTINUOUS SECURITY USE CASE DIAGRAM.231

PART III. CONTINUOUS SLA

FIGURE 1-1, DEVOPS LEMNISCATE.	237
FIGURE 2-1, ROADMAP TO VALUE MODEL.	239
FIGURE 2-2, CONTINUOUS SLA MODEL.	240
FIGURE 2-3, THE ROLE OF SLA CONTROLS IN AN AGILE PROJECT – SLA CONTROL MODEL.	241
FIGURE 2-4, THE ARCHIMATE DESIGN FOR THE BVS.	241
FIGURE 2-5, THE ARCHIMATE DESIGN FOR THE SVS.	242
FIGURE 2-6, THE ARCHIMATE DESIGN FOR THE DVS.	242
FIGURE 2-7, THE INTEGRATED VALUE SYSTEMS BVS, DVS, SVS / ISVS.	243
FIGURE 2-8, APPLICATION SCOPE CONTINUOUS SLA.	243
FIGURE 2-9, BALANCED SCORE CARD [KAPLAN 2004].	244
FIGURE 2-10, ENTERPRISE ARCHITECTURE.	245
FIGURE 2-11, ROADMAP.	246
FIGURE 2-12, VALUE CHAIN OF PORTER, SOURCE: [PORTER 1998].	247
FIGURE 2-13, VALUE STREAM TEMPLATE.	248
FIGURE 2-14, RECURSIVE VALUE CHAIN OF PORTER, SOURCE: [PORTER 1998].	248
FIGURE 2-15, RECURSIVE VALUE CHAIN OF PORTER, SOURCE: [PORTER 1998].	249
FIGURE 4-1, CHANGE PARADIGM.	255
FIGURE 4-2, CHANGE PARADIGM - VISION.	256
FIGURE 4-3, CHANGE PARADIGM - POWER.	258
FIGURE 4-4, CHANGE PARADIGM - ORGANISATION.	260
FIGURE 4-5, CHANGE PARADIGM - RESOURCES.	261
FIGURE 5-1, ROADMAP TO VALUE, BRON: [LAYTON 2017].	266
FIGURE 5-2, CONTINUOUS SLA MODEL.	267
FIGURE 5-3, VALUE STREAM MAPPING MODEL.	268
FIGURE 6-1, CONTINUOUS SLA VALUE STREAM.	271
FIGURE 6-2, USE CASE DIAGRAM FOR THE SET-UP OF CONTINUOUS SLA.	271
FIGURE 7-1, CONTINUOUS SLA MODEL.	277
FIGURE 7-2, RISICO SOURCES.	278
FIGURE 8-1, POSITIONING IN THE ROADMAP TO VALUE.	281
FIGURE 8-2, VALUE CHAIN OF PORTER, BRON: [BOEK MICHAEL PORTER].	281
FIGURE 9-1, SWOT-ANALYSIS.	283
FIGURE 9-2, VALUE STREAM GOVERNANCE MODEL.	284
FIGURE 9-3, CMM MATURITY GOALS.	284
FIGURE 10-1, POSITIONING IN THE ROADMAP TO VALUE.	285
FIGURE 10-2, VALUE STREAM MAPPING MODEL.	286
FIGURE 11-1, POSITIONING IN THE ROADMAP TO VALUE.	287
FIGURE 11-2, VALUE STREAM CANVAS TEMPLATE.	288
FIGURE 11-3, VALUE STREAM CANVAS FOR POLICY ADMINISTRATION.	290
FIGURE 11-4, EXAMPLE CORE VALUE STREAM MAPPING OF A COFFEE MACHINE.	291
FIGURE 11-5, EXAMPLE OF INFORMATION SYSTEM BUILDING BLOCKS OF A COFFEE SERVICE.	291
FIGURE 11-6, SAMPLE APPLICATION SYSTEM BUILDING BLOCKS OF A COFFEE SERVICE.	292
FIGURE 11-7, EXAMPLE OF TECHNOLOGY SYSTEM BUILDING BLOCKS OF A COFFEE SERVICE.	292
FIGURE 11-8, EXAMPLE OF AN SBB-A PICTURE WITH SCOPE / RISK COLORING.	293
FIGURE 13-1, CONTINUOUS MONITORING LAYER MODEL.	297
FIGURE 13-2, MONITOR CLASSIFICATION MODEL.	298
FIGURE 14-1, CONTINUOUS SLA VALUE STREAM.	299
FIGURE 14-2, VALUE CHAIN FROM PORTER FOR THE PERSONALISED COFFEE MACHINE.	300
FIGURE 14-3, SWOT ANALYSIS.	302
FIGURE 14-4, VALUE STREAM MAPPING PERSONALISED COFFEE SERVICE – FUTURE STATE.	304
FIGURE 14-5, VALUE STREAM CANVAS.	305

FIGURE 14-6, INFORMATION BUILDING BLOCKS PICTURE OF THE PERSONALISED COFFEE SERVICE.306
 FIGURE 14-7, APPLICATION BUILDING BLOCKS PICTURE OF THE PERSONALISED COFFEE SERVICE.306
 FIGURE 14-8, TECHNOLOGY BUILDING BLOCKS PICTURE OF THE PERSONALISED COFFEE SERVICE.307

PART IV. CONTINUOUS ASSESSMENT

FIGURE 4-1, CHANGE PARADIGM.321
 FIGURE 4-2, CHANGE PARADIGM – VISION.322
 FIGURE 4-3, CHANGE PARADIGM - POWER.323
 FIGURE 4-4, CHANGE PARADIGM - ORGANISATION.326
 FIGURE 4-5, CHANGE PARADIGM - RESOURCES.327
 FIGURE 5-1, NECKER CUBE.332
 FIGURE 5-2, FRONT SIDE DEVOPS CUBE.333
 FIGURE 5-3, REAR SIDE DEVOPS CUBE.333
 FIGURE 5-4, DEVOPS CE-SPIDER MODEL.336
 FIGURE 6-1, CONTINUOUS DESIGN VALUE STREAM.337
 FIGURE 6-2, USE CASE DIAGRAM FOR CONTINUOUS ASSESSMENT.338
 FIGURE 8-1, DEVOPS CP-SPIDER MODEL.353
 FIGURE 8-2, DEVOPS CN-SPIDER MODEL.356
 FIGURE 8-3, DEVOPS CT-SPIDER MODEL.359
 FIGURE 8-4, DEVOPS CI-SPIDER MODEL.362
 FIGURE 8-5, DEVOPS CD-SPIDER MODEL.365
 FIGURE 8-6, DEVOPS CM-SPIDER MODEL.368
 FIGURE 8-7, DEVOPS CL-SPIDER MODEL.371
 FIGURE 8-8, DEVOPS CY-SPIDER MODEL.374
 FIGURE 8-9, DEVOPS CO-SPIDER MODEL.381
 FIGURE 8-10, DEVOPS CA-SPIDER MODEL.383
 FIGURE 8-11, DEVOPS CQ-SPIDER MODEL.386

Tables

TABLE 1-1, CONTINUOUS EVERYTHING ASPECTS.2
 TABLE 1-2, APPENDICES.4

PART I. CONTINUOUS AUDITING

TABLE 3-1, COMMON PROBLEMS WHEN HANDLING CONTINUOUS AUDITING.18
 TABLE 6-1, USE CASE TEMPLATE.41
 TABLE 6-2, USE CASE FOR CONTINUOUS AUDITING.43
 TABLE 7-1, QUESTIONS TO BE ANSWERED FOR THE STEP 'DETERMINE SCOPE TOM'.46
 TABLE 7-2, MODELS FOR ANSWERS TO THE QUESTIONS OF THE STEP 'DETERMINE SCOPE TOM'.46
 TABLE 7-3, ANSWERS FOR QUESTIONS OF THE STEP 'DETERMINE SCOPE TOM'.47
 TABLE 7-4, QUESTIONS TO BE ANSWERED FOR THE 'SELECT TARGETS' STEP.49
 TABLE 7-5, MODELS FOR ANSWERS TO THE QUESTIONS OF THE 'SELECT TARGETS' STEP.50
 TABLE 7-6, ANSWERS TO QUESTIONS FROM THE 'SELECT TARGETS' STEP.50
 TABLE 7-7, INVENTORY OF EXTERNAL FRAMEWORK OF STANDARDS.53
 TABLE 7-8, QUESTIONS TO BE ANSWERED FOR THE 'IDENTIFY RISKS' STEP.54
 TABLE 7-9, MODELS FOR ANSWERS TO THE QUESTIONS OF THE 'IDENTIFY RISKS' STEP.54
 TABLE 7-10, ANSWERS TO QUESTIONS FROM THE 'IDENTIFY RISKS' STEP.55

TABLE 7-11, CIA-RISK CONTROL MATRIX - OBJECTS.	56
TABLE 7-12, CIA-RISK CONTROL MATRIX - RISKS.....	57
TABLE 7-13, CIA-RISK CONTROL MATRIX - CONTROLS.....	59
TABLE 7-14, QUESTIONS TO BE ANSWERED FOR THE 'REALISE CONTROLS' STEP.	60
TABLE 7-15, MODELS FOR ANSWERS TO THE QUESTIONS OF THE 'REALISE CONTROLS' STEP.....	60
TABLE 7-16, ANSWERS TO QUESTIONS FROM THE 'REALISE CONTROLS' STEP.....	61
TABLE 7-17, QUESTIONS TO ANSWER FOR THE 'MONITOR CONTROLS' STEP.....	62
TABLE 7-18, MODELS FOR ANSWERS TO THE QUESTIONS OF THE 'MONITOR CONTROLS' STEP.....	62
TABLE 7-19, ANSWERS TO QUESTIONS FROM THE 'MONITOR CONTROLS' STEP.	63
TABLE 7-20, QUESTIONS TO BE ANSWERED FOR THE STEP 'DEMONSTRATE EVIDENCE CONTROLS'.	65
TABLE 7-21, MODELS FOR ANSWERS TO THE QUESTIONS OF THE STEP 'DEMONSTRATE EVIDENCE CONTROLS'. ...	65
TABLE 7-22, ANSWERS TO QUESTIONS FROM THE STEP 'DEMONSTRATE EVIDENCE CONTROLS'.	65
TABLE 9-1, MAKE OR BUY A CA-TOOL.....	78
TABLE 9-2, USE CASE FOR CONTROL EVIDENCE MANAGEMENT.	82
TABLE 10-1, CA-TOOL FUNCTIONS.	86

PART II. CONTINUOUS SECURITY

TABLE 3-1, COMMON PROBLEMS WHEN USING CONTINUOUS SECURITY.	108
TABLE 6-1, TERMS PER ISVS USE CASE.	136
TABLE 6-2, USE CASE TEMPLATE.....	138
TABLE 6-3, USE CASE FOR CONTINUOUS SECURITY.	142
TABLE 8-1, USE CASE 'GET TOP MANAGEMENT COMMITMENT'.	147
TABLE 8-2, EXAMPLE OF A STATEMENT OF COMMITMENT.	147
TABLE 8-3, DETERMINE INTERESTED PARTY.	149
TABLE 8-4, INTERESTED PARTIES - INFLUENCE AND INTEREST.	150
TABLE 8-5, INTERESTED PARTIES – REGISTER.....	150
TABLE 8-6, INTERESTED PARTIES – REGISTER EXPLANATION	151
TABLE 8-7, SCOPE - USE CASE.	152
TABLE 8-8, SCOPE - DEFINITION.	153
TABLE 8-9, SCOPE - USE CASE.	154
TABLE 8-10, GOALS.	155
TABLE 8-11, INFORMATION SECURITY POLICY - USE CASE	156
TABLE 8-12, CODE OF CONDUCT EXAMPLE FOR BUSINESS AND IT.	157
TABLE 8-13, CODE OF CONDUCT EXAMPLE FOR IT.....	157
TABLE 9-1, USE CASE 'BEPAL INTERNAL ISSUES'.....	161
TABLE 9-2, EXAMPLES OF INTERNAL ISSUES FACTORS.	162
TABLE 9-3, EXAMPLES VAN IPOPS FACTORS.	163
TABLE 9-4, EXAMPLE IPOPS CLASSIFICATION TEMPLATE.....	163
TABLE 9-5, USE CASE 'DETERMINE EXTERNAL ISSUE'.	165
TABLE 9-6, EXAMPLES OF EXTERNAL ISSUES.	166
TABLE 9-7, EXAMPLES VAN PESTLE FACTORS.....	166
TABLE 9-8, EXAMPLE PESTLE CLASSIFICATION TEMPLATE.....	167
TABLE 9-9, USE CASE 'DETERMINE CRAMM ISSUE'.	169
TABLE 9-10, EXAMPLES OF CRAMM THREATS.	170
TABLE 9-11, TEMPLATE CRAMM-ANALYSE.	171
TABLE 9-12, USE CASE 'DETERMINE RISK CRITERIA'.	174
TABLE 9-13, RISK PRIORITY CRITERIA.....	175
TABLE 9-14, RISK PRIORITY CRITERIA.....	176
TABLE 9-15, RISK PRIORITY CRITERIA.....	176

TABLE 9-16, INTERNAL AUDIT FINDING CRITERIA.	177
TABLE 9-17, THE IMPACT CODE OF INFORMATION SECURITY INCIDENTS.	178
TABLE 9-18, THE IMPACT CODE OF INFORMATION SECURITY INCIDENTS.	180
TABLE 9-19, INFORMATION SECURITY INCIDENT PRIORITY TABLE.	180
TABLE 9-20, INFORMATION SECURITY INCIDENT EVIDENCE MATRIX.	181
TABLE 9-21, USE CASE 'DETERMINE INFORMATION SECURITY ASSETS'.	182
TABLE 9-22, USE CASE 'IDENTIFY RISKS'.	184
TABLE 9-23, IDENTIFIED RISK.	185
TABLE 9-24, TEMPLATE RISK IDENTIFICATION.	186
TABLE 9-25, EXPLANATION OF RISK IDENTIFICATION ITEMS.	186
TABLE 9-26, USE CASE 'IDENTIFY RISKS'.	187
TABLE 9-27, TEMPLATE RISK IDENTIFICATION.	188
TABLE 9-28, TEMPLATE RISK ASSESSMENT.	188
TABLE 9-29, TEMPLATE RISK ASSESSMENT.	188
TABLE 9-30, USE CASE 'RISK TREATMENT OPTIONS RISKS'.	190
TABLE 9-31, MASR TREATMENT OPTIONS.	190
TABLE 9-32, EXPLANATION TREATMENT OPTIONS.	191
TABLE 9-33, USE CASE 'ASSIGNING RISK CONTROLS TO RISKS'.	192
TABLE 9-34, MASR TREATMENT OPTIONS.	193
TABLE 9-35, USE CASE 'ASSIGNING RISK CONTROLS TO RISKS'.	195
TABLE 9-36, USE CASE 'DRAWING UP A RISK TREATMENT PLAN FOR A CONTROL'.	196
TABLE 9-37, TREATMENT PLAN TEMPLATE.	197
TABLE 10-1, USE CASE 'MONITOR EFFECTIVENESS CONTROLS'.	201
TABLE 10-2, USE CASE 'INTERNAL AUDIT PLANNING'.	204
TABLE 10-3, TEMPLATE INTERNAL AUDIT.	205
TABLE 10-4, USE CASE 'INTERNAL AUDIT CRITERIA'.	206
TABLE 10-5, USE CASE 'PERFORMANCE INTERNAL AUDIT'.	208
TABLE 10-6, USE CASE 'REPORT INTERNAL AUDIT'.	209
TABLE 10-7, USE CASE 'INFORMATION SECURITY INCIDENTS'.	211
TABLE 10-8, USE CASE 'INFORMATION SECURITY NC'.	212
TABLE 10-9, USE CASE 'INFORMATION SECURITY NC'.	213
TABLE 10-10, CSI REGISTER EXAMPLE.	214
TABLE 11-1, EFFECTIVENESS ASPECTS OF AGILE SYSTEM DEVELOPMENT.	217
TABLE 11-2, EFFICIENCY ASPECTS OF AGILE SYSTEM DEVELOPMENT.	217
TABLE 11-3, CONTINUOUS SECURITY DEPICTED ON AGILE SCRUM.	223
TABLE 11-4, CONTINUOUS SECURITY MAPPED TO AGILE SCRUM ARTIFACTS.	225
TABLE 11-5, CONTINUOUS SECURITY MAPPED TO AGILE SCRUM EVENTS.	226
TABLE 12-1, CONTINUOUS EVERYTHING ASPECTS.	228
TABLE 12-2, CONTINUOUS SECURITY DEPICTED ON DEVOPS.	232

PART III. CONTINUOUS SLA

TABLE 1-1, CONTINUOUS EVERYTHING ASPECTS.	237
TABLE 2-1, PLANNING OBJECTS.	247
TABLE 3-1, COMMON PROBLEMS WHEN HANDLING CONTINUOUS SLAS.	252
TABLE 6-1, USE CASE TEMPLATE.	272
TABLE 6-2, USE CASE FOR CONTINUOUS SLA.	275
TABLE 7-1, RASCI TABLE FOR THE CONTINUOUS SLA VALUE STREAM.	279
TABLE 8-1, CORE VALUE STREAM EXAMPLES.	282
TABLE 11-1, EXAMPLES OF RISKS AND CONTROLS.	294

TABLE 12-1, EXAMPLES OF RISKS AND CONTROLS.	295
TABLE 14-1, IMPACT OF VISION STATEMENT ON VALUE STREAMS.	300
TABLE 14-2, RISKS BASED ON PORTER'S VALUE CHAINS.	301
TABLE 14-3, RISKS BASED ON THE SWOT.	302
TABLE 14-4, VALUE STREAM GOALS.	303
TABLE 14-5, RISKS BASED ON VALUE STREAM GOALS.	304
TABLE 14-6, RISKS BASED ON SBB-I RISK SESSIONS.	307
TABLE 14-7, RISKS BASED ON SBB-A RISK SESSIONS.	307
TABLE 14-8, RISKS BASED ON SBB-T RISK SESSIONS.	308
TABLE 14-9, EXAMPLES OF RISKS AND CONTROLS.	308
TABLE 14-10, EXAMPLES OF MONITORING RISKS AND CONTROLS.	309

PART IV. CONTINUOUS ASSESSMENT

TABLE 3-1, COMMON ISSUES IN DEVOPS MATURITY.	320
TABLE 5-1, CE MATURITY MODEL.	334
TABLE 5-2, CONTINUOUS EVERYTHING.	335
TABLE 5-3, CMMI LEVELS FOR CONTINUOUS EVERYTHING.	335
TABLE 5-4, PRINCIPLE OF MATURITY LEVELS.	336
TABLE 6-1, USE CASE TEMPLATE.	339
TABLE 6-2, USE CASE FOR CONTINUOUS ASSESSMENT.	342
TABLE 7-1, SIDE 1 QUESTIONS.	344
TABLE 7-2, SIDE 2 QUESTIONS.	345
TABLE 7-3, SIDE 3 QUESTIONS.	346
TABLE 7-4, SIDE 4 QUESTIONS.	347
TABLE 7-5, SIDE 5 QUESTIONS.	348
TABLE 7-6, SIDE 6 QUESTIONS.	349
TABLE 8-1, CP MATURITY CHARACTERISTICS.	353
TABLE 8-2, CN MATURITY CHARACTERISTICS.	355
TABLE 8-3, CT MATURITY CHARACTERISTICS.	359
TABLE 8-4, CI MATURITY CHARACTERISTICS.	361
TABLE 8-5, CD MATURITY CHARACTERISTICS.	364
TABLE 8-6, CM MATURITY CHARACTERISTICS.	367
TABLE 8-7, CL MATURITY CHARACTERISTICS.	370
TABLE 8-8, CY MATURITY CHARACTERISTICS.	373
TABLE 8-9, CQ MATURITY CHARACTERISTICS.	385

Appendices

APPENDIX A, LITERATURE LIST.	389
APPENDIX B, GLOSSARY.	393
APPENDIX C, ABBREVIATIONS.	409
APPENDIX D, WEBSITES.	413
APPENDIX E, INDEX.	415

Introduction

Development & Operations, in short DevOps, has been the starting point for deepening our knowledge of Continuous Everything. This is with reference to the concepts of Continuous Integration/Continuous Deployment (CI/CD) that are frequently discussed in the concept of DevOps. The aspects of DevOps are related to the concept of Continuous and the steps in the development/management cycle (also known as the DevOps Lemniscate).

Understanding DevOps keeps companies busy to provide an optimal interpretation of the 'old' concepts of development and management. Unfortunately, no unambiguous elaboration of DevOps can be found in the literature or on the big Internet. It quickly becomes apparent that DevOps is 'a philosophy'. In other words: not strictly defined and explainable and interpreted in several ways. Companies therefore struggle with this concept. The Concept Continuous Everything gives a simple and uniform structure to define the knowledge and knowhow of each Continuous Everything aspect like Continuous Integration and Continuous Deployment.

This book 'Continuous Control' comprises four aspects of Continuous Everything, namely Continuous Auditing, Continuous Security, Continuous SLA and Continuous Assessment. This forms a large book, in which a piece of knowledge and experience that Bart de Best has gathered in the field of Continuous Control is disclosed.

This book contains a very detailed description of these Continuous Everything aspects of DevOps. This includes the various best practices that are put forward from practical experience in a theoretical context. This context makes it possible to relate the aspects to each other Continuous Everything aspects.

We are proud to have supported Bart with a small group of professionals in the development of all aspects of Continuous Everything. With Bart's unstoppable drive, there is now a very full toolbox with best practices for DevOps. In particular the coherence is a solid addition to the use of the concepts surrounding the aspects of DevOps. Happy reading, flipping through the book, contemplating Continuous Everything!

Dr. Louis van Hemmen – BitAll b.v.

Preface

This book has been compiled from my experiences with Continuous Everything. This concept indicates two aspects of DevOps (Development & Operations), namely Continuous and Everything. The continuous nature of DevOps is mainly reflected in the high frequency of delivery and the fast feedback that is obtained as a result. Everything refers to the fact that not only software must be delivered Continuously, but that all aspects of computerisation must move along with it.

This book focuses on Continuous Control. The goal of this Continuous Everything area is to control all the identified risks of the information provisioning in order to safeguard the realisation of the intended outcome improvement of the business value streams and thus achieve the business goals. This book is a bundle of four CE books namely Continuous Auditing, Continuous Security, Continuous SLA and Continuous Assessment.

This is a snapshot of the best practices I am using right now. Given 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 on how to deal with this area of Continuous Everything, I have decided to keep this book Agile. This means that in this book I briefly describe every aspect. I hereby share important insights that I have gained during my role as a consultant, trainer, coach, and examiner with regard to related work of this area. Where appropriate, I refer to sources that I myself have consulted for further training. I realise that these best practices will not apply to all information systems and that the approach is a snapshot that may be outdated due to the increasing speed of innovation.

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

I would like to express my sincere thanks to the following people for their inspiring contribution to this book and the great collaboration!

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

I wish you a lot of fun reading this book and, above all, much success in applying these aspects of Continuous Everything within your own organisation.

If you have any questions or comments, please don't hesitate to contact me. A lot of time has gone into making this book as complete and consistent as possible. Should you nevertheless find shortcomings, I would appreciate it if you would inform me, so that these matters can be incorporated in the next edition.

Bart de Best, Zoetermeer.
bartb@dbmetrics.nl.

1 Introduction

Reading guide:

The first section of this chapter sets out the purpose of this book (1.1). Then the target group (1.2) is named. Section 1.3 discusses the background of Continuous Control and section 1.4 the structure and content of the book by briefly stating what is covered by each part. This chapter concludes with a reading guide (1.5).

1.1 Objective

The primary objective of this book is to provide a Continuous Control toolbox. This book discusses four key Continuous Control aspect areas. There are certainly many other aspect areas of Continuous Control, but the ones selected in this book are a good foundation. The depth of the aspect areas has been kept limited due to the limited publication space. The book is intended as a reference for anyone involved with DevOps.

1.2 Target group

The target group of Continuous Control are all involved officials in the DevOps teams. This includes the architects, 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 line managers, process owners, process managers, etcetera who are involved in the creation of information provision through a DevOps method. Finally, there is a target group that does not develop or manage, but that determines whether the value streams meet the required criteria. This target group includes quality employees and auditors. They can use this book to identify risks that need to be accepted or controlled.

1.3 Background

This book contains various methods and techniques to give content to Continuous Control in a continuous way. The DevOps Lemniscate provides an overview of the key Continuous Everything aspect areas, as shown in Figure 1-1. Though Continuous Auditing, Continuous Security, Continuous SLA and Continuous Assessment are not plotted in the DevOps Lemniscate. However the CE aspects that are depicted in the DevOps Lemniscate are under control of these four Continuous Control aspects.

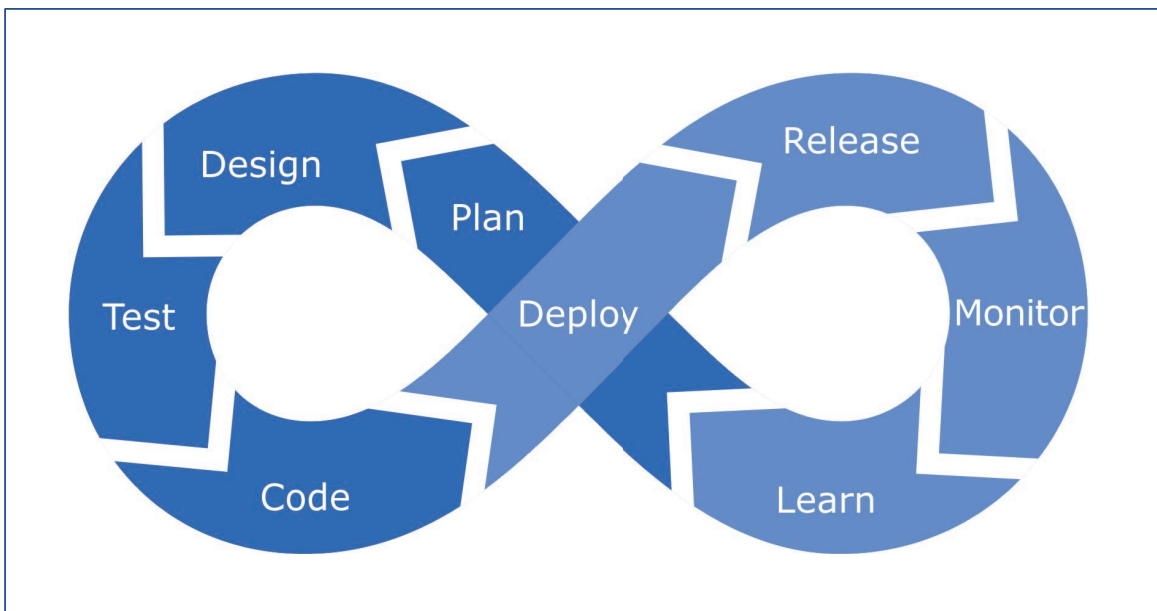


Figure 1-1, DevOps Lemniscate.

The DevOps lemniscate provides an overview of the phases to be followed to continuously produce software. The DevOps Lemniscate is therefore a good basis for defining the concept of Continuous Everything (CE). The four Continuous Control aspects are integrated into all steps in the DevOps Lemniscate.

The CE concept describes all phases of the DevOps Lemniscate in the form of activities to be performed continuously. 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)	7	Continuous Monitoring (Monitor)
2	Continuous Design (Design)	8	Continuous Learning (Learn)
3	Continuous Testing (Test)	9	Continuous Auditing (-)
4	Continuous Integration (Code)	10	Continuous Security (-)
5	Continuous Deployment (Deploy)	11	Continuous SLA (-)
6	Continuous Deployment (Release)	12	Continuous Assessment (-)

Table 1-1, Continuous Everything aspects.

Continuous Auditing (9), Continuous Security (10), Continuous SLA (11) and Continuous Assessment (12) are not represented in the DevOps Lemniscate, as are other Continuous aspect areas such as Continuous Robotics and Continuous Growth. This is done to keep the DevOps Lemniscate uncomplicated.

The word 'Continuous' refers to a number of characteristics that indicate the work within a DevOps team. Firstly, the frequency of actions is higher than in traditional system development. This relates to both the construction and the deployment of what has been built. This can vary from minutes, hours, and days in deployment frequency. In addition, 'Continuous' refers to a holistic view of the work. For example, monitoring is not limited to the production environment, but all environments are monitored. Also, not only the products and services are monitored, but also value streams and even people's knowledge and skills. This is in line with the people, process, partner, and technology views of ITIL 4. Finally, the term 'Continuous' indicates that all phases of the DevOps Lemniscate are related to each other. For example, Continuous Testing is used in the steps 'Plan', 'Design', 'Code', 'Deploy' and 'Monitor'.

1.4 Structure

This book has been constructed by summarising four previously published books in this book, namely:

- part I DevOps Continuous Auditing
- part II DevOps Continuous Security
- part III DevOps Continuous SLA
- part IV DevOps Continuous Assessment

1.4.1 Part I, Continuous Auditing

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.

The content of this part of the book is 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.

1.4.2 Part II, Continuous Security

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

- The business perspective which focuses on the business value streams that must be in control of the identified risks by continuously testing the effectiveness of the controls deployed and recording evidence.
- The development perspective which focuses on the development value streams that must be in control by integrally including the non-functional requirements for information security in the development.

- The operations perspective which focuses on the operations value streams that must be in control of 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 requirements.

The content of this part of the book 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.

1.4.3 Part III, Continuous SLA

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.

The content of this part of the book 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.

1.4.4 Part IV, Continuous Assessment

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. Continuous Assessment provides a tool to make the DevOps teams aware of where they stand in terms of development and which next steps they can take to develop.

Continuous Assessment consists of two assessment models and assessment questionnaires. The first model is the DevOps Cube model which is based on the idea that DevOps can be viewed from six different perspectives of a cube, namely: 'Flow', 'Feedback' and 'Continuous Learning', 'Governance', 'Pipeline' and 'QA'. The second model is the DevOps CE model which is based on the Continuous Everything perspectives: 'Continuous Planning', 'Continuous Design', 'Continuous Testing', 'Continuous Integration', 'Continuous Deployment', 'Continuous Monitoring', 'Continuous Learning' and 'Continuous Security'. And in addition the assessments for 'Continuous Documentation' and 'Continuous Auditing' is included.

1.5 Appendices

The appendices contain important information that helps to better understand Continuous Everything.

Appendix	Subject	Explanation
A	Literature references	In this book reference is made to consulted literature in the form of: [Author Year]. In the appendix, the full name of the author, the title and the ISBN number are given.
B	Glossary	Only the main concepts are explained in this appendix.
C	Abbreviations	Within the world of DevOps many abbreviations are used. Frequently used terms have been abbreviated for the readability of this book. The first time an abbreviation is used, it is shown in brackets behind the full term.

Appendix	Subject	Explanation
D	Websites	A number of relevant websites are included in this appendix. In this book, these websites are referred to 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 in this book is limited. However, terms that keep coming back are represented 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 2022 CD	B. de Best, "Continuous Deployment", English language, Leonon Media, 2022, ISBN13: 978 94 92618 733.
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.
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.

References	Publications
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.
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.

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

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) proves that high-performing organisations using DevOps practices are outperforming their non-high performing peers in many following areas [Kim 2016].
Canary releasing pattern	Normally a release is offered to every user at once. Canary releasing is the approach in which a small set of users is receiving the new release. If this small scope release works fine than the release can be deployed to all users. The term canary refers to the old habit to have a canary in the coal mines to detect toxic gas.
Change categories	Changes can be categorised into standard changes, normal changes and urgent changes.
Change schedules	Changes can be scheduled in order to defined in which order they have to be applied.
Cloud configuration files	Cloud configuration files are used to initiate a cloud service before using it. In this way cloud service providers enable customers to configure the cloud environment for their needs.
Cluster immune system release pattern	The cluster immune system expands upon the <u>canary release pattern</u> by linking our production monitoring system with our release process and by automating the roll back of code when the user-facing performance of the production system deviates outside of a predefined expected range, such as when the conversion rates for new users drops below our historical norms of 15%–20% [Kim 2016].
Code branch	See <u>branching</u> .
Code review methods	Code review can be performed in several ways like “ <u>over the shoulder</u> ”, <u>pair programming</u> , <u>email pass-around</u> and <u>tool-assisted code review</u> .
Codified NFR	A list of Non-Functional Requirements (NFR) that are categorised in categories like availability, capacity, security, continuity et cetera.
Collaboration	One of the four pillars of DevOps is collaboration. Collaboration refers to the way the individuals of a DevOps team works together to achieve the common goal. There are many forms in which this collaboration comes to expression like: <ul style="list-style-type: none"> • peer to peer programming; • demonstrating weekly progress; • documentation; et cetera.
Commit code	Committing code is the action in which the DevOps engineer adds the changed source code to the repository, making these changes part of the head revision of the repository [Wiki].
Commit stage	This is the phase in the CI/CD secure pipeline where the source code is compiled to the object code. This includes the performance of the unit testcases.
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.
Compliance officer	The compliance officer is a DevOps role. The compliance officer is responsible for ensuring compliance with agreed standards throughout the whole life cycle of a product.
Configuration management	Configuration Management refers to the process by which all artefacts, and the relationships between them, are stored, retrieved, uniquely identified and modified.
Containers	A container is an isolated structure that is used by DevOps engineers to build their application independently from the underlying operating system or hardware. This is accomplished by interfaces in the container that are used by DevOps engineers. Instead of installing the application in an environment, the complete container is deployed. This saves a lot of dependencies and prevents configuration errors to occur.
Conway's law	The following statement of Melvin Conway is called the Conway's law: "organisations which design systems ... are constrained to produce designs which are copies of the communication structures of these organisations." [Wiki].
Cultural debt	There are three forms of debt. Cultural debt, <u>technical debt</u> and <u>information debt</u> . This form of debt refers to the decision to keep flaws in the organisation structure, hiring strategy, values et cetera. This debt costs interest and will result in less maturity growth of the DevOps teams. Cultural debt can be recognised by the exitance of extensive silos, workflow constraints, miscommunications, waste et cetera.
Culture, Automation Measurement, Sharing (CAMS)	<p>CAMS is the abbreviation for Culture, Automation, Measurement and Sharing.</p> <ul style="list-style-type: none"> • Culture: Culture relates to the people and process aspects of DevOps. Without the right culture, automation attempts will be fruitless. • Automation: Release management, configuration management, and monitoring and control tools should enable automation. • Measurement: 'If you can't measure it, you can't manage it.' & 'If you can't measure it, you can't improve it'. • Sharing: Culture of sharing ideas and problems is critical to help organisations to improve. Creates feedback loop.
Cycle time (flow time)	Cycle time measures more the completion rate or the work capability of a system overall, and a shorter cycle time means that less time is being wasted when a request has been made but no progress or work is getting done.
Cycle time (lean)	The average time between two successive units leaving the work or manufacturing process.
Declarative programming	This is a <u>programming paradigm</u> that expresses the logic of a computation without describing its control flow. An example are the database query languages for example TSQL and PSQL.

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. <ul style="list-style-type: none"> • The first act begins in IT Operations where technical debt results in jeopardising our most important organisational promises. • The second act starts with compensating the latest broken promise by promising a bigger, bolder feature or an even larger revenue target. As a result, Development is tasked with another urgent project which results in even more technical debt. • The third stage is where the deployments are getting slower and slower, and outages are increasing. The business value continuously decreases.
E-mail pass-around	E-mail pass-around is a review technique where the source code management system emails code to reviewers automatically after the code is checked in [Kim 2016].
Error path	See <u>happy path</u> .
Fast feedback	Fast feedback refers to the second way of the three ways of Gene Kim. The second way is about having feedback on the functionality and quality of the product that is created or modified as soon as possible in order to maximise the business value.
Feature toggles	A feature toggle is a mechanism that makes it possible to enable or disable a part of the functionality of an application released in production. Feature toggles enables testing the effect of changes on users in production. Feature Toggles are also referred to as Feature Flags, Feature Bits or Feature Flippers.
Feedback	Feedback within the context of DevOps is the mechanism by which errors in the value stream are detected as soon as possible and is used to improve the product and if necessary to improve the value stream as well.
Feedforward	Feedforward within the context of DevOps is the mechanism by which experiences in the present value stream are used to improve the future value stream. Feed forward is the opposite of feedback since feedback is focused on the past and feed forward on the future.
Gaussian distribution	In probability theory, the normal (or Gaussian) distribution is a very common continuous probability distribution. Normal distributions are important in statistics and are often used in the natural and social sciences to represent real-valued random variables whose distributions are not known. A random variable with a Gaussian distribution is said to be normally distributed and is called a normal deviate [Wiki].

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. <ul style="list-style-type: none"> • Independent: The product backlog item should be self-contained, in a way that there is no inherent dependency on another product backlog item. • Negotiable: Product backlog items, up until they are part of an iteration, can always be changed, rewritten or even discarded. • Valuable: Product backlog item must deliver value to the stakeholders.

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

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

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

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: <ul style="list-style-type: none"> • information management; • and on-the-spot decision making.
One piece flow	The Lean approach means that the DevOps team only works at one item at a time as a team with a fast pace and smooth flow. This is also used in the first way of the three ways of Gene Kim.
Operations	Operations is the team often responsible for maintaining the production environment and helping to ensure that required service levels are met [Kim 2016].
Operations stories	The work that has to be done by Ops can be written in stories. In that way that can be prioritised and managed.
OPS liaison	An OPS liaison is an operation employee who is assigned to a development team in order to facilitate the development team for their infrastructural demands.
Organisation archetypes	There are three organisation archetypes: functional, matrix, and market. They are defined by Dr. Roberto Fernandez as follows: <ul style="list-style-type: none"> • Functional: Functional-oriented organisations optimise for expertise, division of labour, or reducing cost. • Matrix: Matrix-oriented organisations attempt to combine functional and market orientation. • Market: Market-oriented organisations optimise for responding quickly to customer needs.
Organisational typology model	This a model of Dr. Ron Westrum in which he defined three types of culture: 'pathological', 'bureaucratic', 'generative'. These organisation types can be recognised by the following characteristics: <ul style="list-style-type: none"> • Pathological organisations are characterised by large amounts of fear and threat. • Bureaucratic organisations are characterised by rules and processes. • Generative organisations are characterised by actively seeking and sharing information to better enable the organisation to achieve its mission.

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

Term	Meaning
SBAR	<p>This technique offers guidelines for making sure concerns or critiques are expressed in a productive manner.</p> <p>In this situation the people who concerns it have to follow the following steps:</p> <ul style="list-style-type: none"> • situational information to describe what is happening; • background information or context; • an assessment of what they believe the problem is; • recommendations for how to proceed.
Security testing	<p>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.</p>
Self service capability	<p>One way of integrating Ops in Dev is the usage of infrastructure self-services.</p>
Shared goals	<p>Delivering value to the customer requires that Dev and Ops are working together in value streams and have shared goals and practices.</p>
Shared Operations Team (SOT)	<p>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.</p>
Shared version control repository	<p>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.</p>
Simian army	<p>Simian Army consists of services (Monkeys) for generating various kinds of failures, detecting abnormal conditions, and testing the ability to survive them.</p> <p>The goal is to keep the cloud service safe, secure, and highly available. Currently there are 3 Monkeys in the Simian Army:</p> <ul style="list-style-type: none"> • Janitor Monkey (unused resources); • Chaos Monkey (try to shut down a service); • Conformity Monkey (non-conformance to rules).
Single repository	<p>A single repository is used to facilitate trunk-based development.</p>
Smoke testing	<p>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.</p> <p>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.</p>
Standard deviation	<p>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].</p>
Standard operations	<p>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.</p>

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

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

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: <ul style="list-style-type: none"> • Unnecessary software features. • Communication delays. • Slow application response times. • Overbearing bureaucratic processes.
Waste reduction	Minimisation of waste at its source is to minimise the quantity required to be treated and disposed of, achieved usually through better product design and/or process management. Also called waste minimisation [Businessdictionary] .
WIP limit	This is a Key Performance Indicator (KPI) that is used in the Kanban process to maximise the number of items that has been started but that is not completed. Limiting the amount of WIP is an excellent way to increase throughput in your software development pipeline.
Work In Progress (WIP)	Material that has entered the production process but is not yet a finished product.

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
BOK	Body of Knowledge
BSC	Balanced Score Card
BVS	Business Value System
CA	Competitive Advantage
CA	Continuous Auditing
CAB	Change Advisory Board
CAMS	Culture, Automation, Measurement and Sharing
CD	Continuous Deployment
CE	Continuous Everything
CEM	Central Event Monitor
CEMLI	Configuration, Extension, Modification, Localisation, Integration
CEO	Chief Executive Officer
CFO	Chief Finance Officer
CI	Configuration Item
CI	Continuous Integration
CIA	Confidentiality, Integrity & Accessibility (or Availability)
CIO	Chief Information Officer
CL	Continuous Learning
CM	Continuous Monitoring
CMDB	Configuration Management DataBase
CMMI	Capability Maturity Model Integration
CMS	Configuration Management System
CN	Continuous design
CO	Continuous dOcumentation
CoC	Code of Conduct
CoP	Communities of Practice
CP	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
CT	Continuous Testing
CTO	Chief Technical Officer

Abbreviation	Meaning
CY	Continuous security
DevOps	Development & Operations
DML	Definitive Media Library
DNS	Domain Name System
DoD	Definition of Done
DoR	Definition of Ready
DTAP	Development, Test, Acceptance and Production
DU	Definitional Uncertainty
DVS	Development Value System
E2E	End-to-End
ERD	Entity Relation Diagram
ERP	Enterprise Resource Planning
ESA	Epic Solution Approach
ESB	Enterprise Service Bus
ETL	Extract Transform & Load
EUX	End User eXperience Monitoring
FAT	Functionele AcceptatieTest
FSA	Feature Solution Approach
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
JIC	Just In Case
JIT	Just In Time
JKK	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
MOF	Microsoft Operations Framework
MRI	Minimum Required Information
MT	Module Test
MTBF	Mean Time Between Failure
MTBSI	Mean Time Between System Incidents
MTTR	Mean Time To Repair
MVP	Minimal Viable Product
NC	Non-Conformity
NFR	Non-Functional Requirement
OAWOW	One Agile Way of Working
OLA	Operational Level Agreement
PAAS	Platform As A Service
PAT	Production Acceptance Test
PBI	Productie Backlog Item
PDCA	Plan Do Check Act
PESTLE	Political, Economic, Sociological, Technological, Legislative, Environmental
POR	Project or Organisational Risk
PPT	People, Process & Technology
PST	Performance StressTest
PT	Processing Time
QA	Quality Assurance
QC	Quality Control
RACI	Responsibility, Accountable, Consulted, and Informed
RASCI	Responsibility, Accountable, Supporting, Consulted and Informed
RBAC	Role Based Access Control
REST API	REpresentational State Transfer Application Programming Interface
ROI	Return On Investment
RPA	Robotic Process Automation
RUM	Real User Monitoring
S-CI	Software Configuration Item
SA	Strategic IS Architecture
SAFe	Scaled Agile Framework
SAT	Security AcceptatieTest
SBAR	Situation, Background, Assessment, Recommendation

Abbreviation	Meaning
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
TCO	Total Cost of Ownership
TCP	Transmission Control Protocol
TDD	Test Driven Development
TFS	Team Foundation Server
TISO	Technical Information Security Officer
TOM	Target Operating Model
TPS	Toyota Production System
TTM	Time To Market
TU	Technical Uncertainty
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
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-publicaties_2021-07-22_900.pdf
De Caluwé	[De Caluwé]	https://www.agile4all.nl/het-kleurenmodel-van-de-caluwe-en-vermaak/
DevOps	[DevOps]	http://DevOps.com
DDD	[DDD]	https://www.slideshare.net/skillsmatter/ddd-in-agile
doxygen	[doxygen]	http://www.doxygen.nl/manual/docblocks.html
doxygen example	[doxygen example]	http://www.doxygen.nl/manual/examples/qtstyle/html/class_q_tstyle_test.html#a0525f798cda415a94fedeceb806d2c49
EXIN	[Exin]	http://www.exin.nl
Gladwell	[GLADWELL]	http://www.gladwill.nl
IIR	[IIR]	http://www.IIR.nl
Investopedia	[Investopedia]	https://www.investopedia.com
ITMG	[ITMG]	http://www.ITMG.nl
ITPedia	[ITPEDIA]	http://www.itpedia.nl
Patrick Cousot	[Patrick Cousot]	https://www.di.ens.fr/~cousot/abstract_interpret.shtml
Porter	[Porter]	https://medium.com/@sniloy/value-chain-analysis-value-stream-mapping-and-business-process-mapping-what-is-the-difference-431589d27ea8
Sneider	[Schneider]	https://shift314.com/are-you-using-the-right-culture-model/
Tiobe	[Tiobe]	www.tiobe.com/content/paperinfo/DefinitionOfConfidenceFactor.html
UnitTest	[UnitTest]	https://docs.python.org/3/library/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_documentation_generators

Table D-1, Websites.

Appendix E, Index

%

%C/A · 50, 52, 100, 146, 148, 151, 153, 155, 160, 164, 165, 167, 168, 173, 181, 184, 187, 189, 192, 194, 196, 200, 203, 205, 207, 208, 209, 211, 212, 269, 361, 409

A

A/B testing · 345, 357, 393
 abuse of audit tools · 169
 abuse of information system · 169
 acceptance
 - criteria · 135, 141, 172, 173, 174, 176, 183, 206, 220, 229, 394, 398
 - criterium · 347, 364, 378
 - test · 393
 account · 171, 178
 actor · 40, 41, 42, 80, 138, 140, 146, 149, 151, 153, 154, 155, 156, 160, 165, 168, 169, 173, 181, 182, 184, 187, 189, 192, 194, 196, 200, 203, 205, 206, 207, 208, 209, 212, 213, 272, 273, 274, 339, 340
 ad hoc deployment · 362, 376
 ad hoc monitoring · 365, 376
 ad hoc testing · 356, 376
 adaptive change · 343
 adaptive software development · 218
 added value · 122, 143, 153, 216, 217, 218, 220, 221, 327, 329, 370
 additional change · 343
 affinity · 393
 Agile · 393, 405, 406
 - infrastructure · 393
 - modelling · 218
 - planning · 344
 - process · 349
 - project · 237, 238, 251, 252, 282, 352, 385
 - testing · 345
 - unified process · 218
 Agile Scrum · 60, 95, 96, 102, 105, 114, 115, 117, 133, 215, 218, 219, 220, 222, 223, 224, 225, 226, 227, 324, 325, 346, 382, 397
 Agile Scrum framework · 324
 Agile Scrum process · 114, 215, 346
 Agile way of working · 253
 alternate path · 393
 Amazon Web Services · See AWS
 Andon cord · 393
 anomaly detection technique · 393
 antifragility · 394
 anti-pattern · 112, 113, 116, 117, 119, 322, 323, 325, 327, 393
 application architecture · 245

application component · 401
 application management · 400
 Application Services Library · See ASL
 Archimate · 241, 242
 architecture · 238, 245
 - building block · 344
 - model · 3, 265, 313, 315, 316, 317, 331
 - principle · 238, 263, 313, 329, 330, 331
 artefact · 384, 394, 396
 artefact repository · 394
 ASL · 241, 409
 assessment · 380, 404
 asset · 135, 141, 162, 181, 182, 371, 372
 - category · 135
 - group · 135
 - inventory · 175
 - register · 135, 153, 183
 assisted code review used · 345
 auditability · 363, 378
 automated regression testing · 357
 automated sign off · 357, 375
 automated test · 394
 availability · 155, 177, 178, 179, 180, 361, 366, 395
 AVS · 16
 awareness training · 141, 148, 155, 156
 AWS · 58, 409

B

backlog item · 403
 bad apple theory · 394
 bad path · 358, 394
 Balanced Score Card · 244, 248, See BSC
 baseline · 358, 360
 BDD · 316, 317, 345, 348, 354, 374, 394, 409
 Behavior Driven Development · 357, See BDD
 benchmark · 317, 319
 best practice · 395
 BI · 93, 94, 409
 binary · 394
 BiSL · 241, 409
 blameless post mortem · 346, 375, 394
 blamelessness · 394
 blue/green deployment · 394
 Body of Knowledge · See BOK
 BOK · 409
 bottleneck · 269, 294, 316, 327
 boundary · 50, 53, 287, 313, 354
 branching · 359, 360, 361, 395
 breach · 170
 broken build · 361, 394
 brown field · 394
 BSC · 49, 98, 239, 244, 266, 267, 334, 385, 409

- BSC · 48
 - build · 343, 345, 346, 347, 357, 358, 360, 361, 375, 378, 394, 395, 396, 405
 - automation · 360
 - meta data · 360, 377
 - time · 361
 - build-in failure mode · 346, 361, 378
 - burn down chart · 385
 - burn up chart · 385
 - business
 - case · 107, 111, 116, 156, 229, 230, 319, 321, 331
 - impact · 163, 164, 167
 - process · 100, 217
 - service monitoring · 63
 - value chain · 16, 102, 105
 - Business Information Services Library · See BiSL
 - Business Intelligence · See BI
 - business value · 395, 397
 - Business Value System · See BVS
 - BVS · 102, 103, 123, 128, 239, 241, 243, 248, 249, 251, 252, 256, 263, 264, 267, 273, 409
-
- ## C
- CA · 409
 - CAB · 347, 409
 - CAMS · 396, 409
 - canary release · 363, 395
 - capability · 396
 - Capability Maturity Model Integration · See CMMI
 - capaciteit · 395
 - capacity · 361, 366
 - CCTA Risk Assessment Method Methodology · See CRAMM
 - CD · 237, 334, 359, 364, 365, 374, 375, 376, 377, 378, 395, 399, 409
 - CDAAS · 362, 364, 375
 - CE · 409
 - CE model · 351
 - CEM · 409
 - CEMLI · 343, 409
 - CE-model · 351
 - Central Event Monitor · See CEM
 - Central Processing Unit · See CPU
 - CEO · 146, 147, 148, 409
 - CFO · 146, 147, 148, 409
 - chain · 247, 248, 249, 313, 324, 325, 335, 347, 355, 358, 359, 361, 364, 367, 370
 - chain management · 361, 376
 - chain manager · 347
 - change
 - authority · 347
 - category · 395
 - manager · 347
 - object · 347
 - paradigm · 21, 95, 111, 112, 113, 114, 117, 118, 121, 229, 238, 255, 256, 257, 259, 261, 263, 313, 321, 322, 323, 326, 327, 329
 - schedule · 395
 - Change Advisory Board · See CAB
 - check-in · 360, 374, 376, 380
 - check-out · 360
 - Chief Executive Officer · See CEO
 - Chief Finance Officer · See CFO
 - Chief Information Officer · See CIO
 - chief product owner · 347
 - Chief Technology Officer · See CTO
 - CI · 237, 334, 361, 362, 364, 374, 375, 376, 377, 378, 395, 399, 409
 - CI/CD secure pipeline · 93, 102, 114, 154, 157, 201, 202, 229, 230, 232, 233, 251, 252, 261, 268, 280, 295, 313, 322, 324, 326, 372
 - CIA · 93, 104, 135, 136, 140, 153, 154, 155, 157, 159, 160, 161, 162, 164, 168, 169, 171, 172, 174, 178, 179, 180, 183, 184, 185, 186, 187, 188, 191, 193, 194, 195, 196, 198, 201, 202, 206, 210, 215, 224, 225, 227, 409
 - CIA-matrix · 104, 136, 153, 155, 157, 159, 168, 172, 183, 184, 186, 191, 194, 195, 196, 198, 201, 202, 206, 215, 227
 - CIO · 409
 - CISO · 146, 147
 - CL · 237, 334, 370, 371, 374, 375, 376, 377, 378, 379, 409
 - clean deployment · 364
 - cloud · 395
 - cloud configuration file · 395
 - cloud provider · 149
 - cloud service · 107, 112, 395
 - cluster immune system release pattern · 395
 - CM · 237, 334, 353, 355, 367, 373, 375, 376, 377, 378, 381, 383, 409
 - CMDB · 62, 362, 375, 409
 - CMM · 284
 - CMMI · 335, 409
 - CMS · 364, 409
 - CN · 237, 409
 - CO · 334, 381, 409
 - CoC · 135, 156, 157, 409
 - code branch · 395
 - Code of Conduct · 141, 155, 156, 157, 158, 162, 181, See CoC
 - code review form · 395
 - Code Tabaksblatt · 50
 - code view · 295, 297, 308
 - codified NFR · 395
 - collaborating tool · 379
 - collaboration · 395
 - commit code · 395
 - commit stage · 395
 - commitment statement · 135, 146, 147
 - Communities of Practice · See CoP
 - competence · 253, 368, 369, 370, 376, 379, 393, 398
 - matrix · 369
 - monitor information · 370, 377

- partnership · 370, 379
- Competitive Advantage · See CA
- Competitive Response · See CR
- Completeness / Accurateness · See %C/A
- compliance · 93, 107, 108, 117, 122, 123, 128, 147, 156, 158, 195, 204, 229, 396
- compliance checking · 396
- compliance officer · 396
- compliance officer · 396
- compliance officer · 396
- component · 399, 402, 405
- confidentiality · 93, 104, 113, 154, 178, 179, 180
- Confidentiality, Integrity & Accessibility · See CIA
- Configuration Item · See CI
- configuration management · 396
- Configuration Management DataBase · See CMDB
- Configuration Management System · See CMS
- Configuration, Extention, Modification, Localisation, Integration · See CEMLI
- consequential incident · 260, 261
- consistency check · 344
- constant pace · 349
- container · 396
- context · 77
- contextual inquiry · 345
- contingency recovery option · 58
- continuity · 361, 395, 401
- continuous
 - assessment · 2, 19, 31, 40, 41, 47, 64, 66, 68, 70, 73, 76, 78, 83, 88, 228, 232, 319, 339
 - auditing · 2, 17, 232
 - auditing concept · 65
 - auditing engine · 65
 - auditing pyramid · 11, 35
 - control model · 127
 - delivery · 344, 349, 362
 - deployment · 2, 228, 232
 - design · 2, 124, 228, 229, 230, 232
 - documentation · 344
 - everything · 335
 - improvement · 406
 - integration · 2, 218, 228, 232, 316, 344
 - learning · 2, 228, 232, 317, 332, 351, 406
 - monitoring · 2, 228, 232, 344
 - monitoring layer model · 297
 - planning · 2, 228, 231
 - security · 94, 107, 136, 137, 138
 - SLA model · 238, 239, 266, 277
 - testing · 2, 228, 232, 344, 358
 - testing roadmap · 18
- Continuous aSessment · 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 · 409, See CP
- Continuous securitY · See CY
- continuous security assessment · 96
- continuous security pyramid · 97, 102, 111, 112, 114, 115, 121, 124, 125, 126, 131, 132, 133, 228
- Continuous SLA · 409, See CQ
- Continuous Testing · 265, See CT
- contract · 216
- control · 93, 94, 95, 97, 98, 99, 102, 103, 104, 105, 107, 108, 109, 112, 113, 114, 115, 116, 117, 118, 122, 123, 124, 126, 127, 128, 130, 132, 133, 139, 142, 143, 147, 152, 153, 159, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 207, 211, 215, 223, 224, 225, 226, 227, 232, 233, 347, 371, 372, 373, 375, 396, 404
 - backlog · 105
 - definition database · 65
 - evidence database · 65, 136
 - lifecycle management · 103
 - requirement · 94
- Conway's law · 396
- cookbook · 163, 167, 172, 186, 188, 191, 193, 197, 202, 205
- CoP · 115, 324, 409
- core value stream · 240, 248, 251, 252, 253, 256, 257, 263, 264, 266, 267, 268, 273, 274, 275, 279, 280, 283, 284, 285, 287, 291, 293, 294, 297, 299, 301, 302, 303
- core value stream mapping · 273, 279, 285, 291
- corporate espionage · 169, 171
- corrective change · 343
- counter measure · 400
- countermeasure · 93, 94, 98, 103, 104, 107, 130, 132, 139, 145, 153, 154, 155, 157, 189, 191, 195, 196, 197, 240, 251, 252, 255, 256, 260, 264, 266, 277, 279, 326, 344, 364
- CP · 237, 239, 263, 409
- CPU · 365, 409
- CQ · 251, 273, 385, 409
- CR · 409
- CRAMM · 409
- CRAMM · 54
- CRAMM issue · 104, 169
- CRAMM issue register · 135
- CRAMM threat · 170
- CRC · 409
- CRC check · 307, 308, 309
- CRC code · 154
- Critical Success Factor · See CSF
- CRM · 282, 293
- CS · 263, 409
- CSF · 135, 145, 153, 154, 155, 157, 160, 251, 264, 283, 409
- CSF-scheme · 51
- CSI register · 136, 214

CT · 237, 334, 358, 359, 374, 375, 376, 377, 378, 409
 CTO · 409
 cultural debt · 396
 Culture, Automation, Measurement and Sharing · See CAMS
 current state · 287
 custom software · 112
 customer · 149, 150, 151
 CY · 410
 cycle time · 396
 Cyclic Redundancy Check · See CRC

D

daily stand-up · 344
 damage · 170
 data
 - analysis tools · 94
 - driven testing · 358, 377
 - leakage · 94
 - loss · 155
 - masking · 358, 377
 debt · 396
 declarative programming · 396
 defect · 347, 358, 361, 364, 401
 - management · 361, 376
 - record · 358, 361, 377
 defect tracking · 397
 Definition of Done · See DoD
 Definition of Ready · See DoR
 Definitional Uncertainty · See DU
 Definitive Media Library · See DML
 Demming wheel · 400
 deployment · 345, 358, 362, 363, 393
 - defect · 364, 377
 - frequency · 2
 - management · 364, 376
 - pipeline · 358, 364, 395
 - script · 363
 - strategy · 363, 378
 - team · 220
 design · 95, 228, 229, 396, 407, 412
 Dev engineer · 1
 Dev profile · 346
 development · 393, 394, 397, 399, 400, 402, 404, 405, 406, 407
 Development & Operations · See DevOps
 development ritual · 397
 Development Value System · See DVS
 Development, Test, Acceptance and Production · See DTAP
 deviation · 97, 117, 208, 344
 DevOps · 43, 81, 93, 94, 95, 96, 101, 102, 111, 114, 115, 116, 118, 119, 125, 133, 227, 228, 229, 230, 231, 232, 371, 374, 383, 389, 393, 395, 397, 403, 407, 410
 - Cube model · 3, 329
 - engineer · 317, 324, 325, 327, 328, 329, 330, 331, 340, 356, 369, 393, 395, 396, 397, 404, 405, 406

- Lemniscate · 1, 2, 93, 95, 125, 228, 231, 237, 313, 316, 319, 329, 351
 - maturity · 261
 - method · 1, 339
 - team · 114, 115, 116, 229, 230, 343, 344, 347, 357, 359, 361, 362, 363, 364, 367, 368, 369, 370, 379, 380, 393, 394, 395, 396, 398, 402, 407
 digitisation · 113, 166
 disaster · 170
 DML · 363, 375, 410
 DML control · 375
 DNS · 410
 document lifecycle management · 380
 documentation coverage · 380
 documentation generating tool · 379
 DoD · 42, 60, 93, 220, 226, 249, 250, 257, 259, 264, 265, 268, 335, 339, 340, 344, 347, 351, 360, 369, 372, 378, 380, 382, 384, 399, 410
 DoD effectiveness · 347
 domain architect · 278
 Domain Name System · See DNS
 DoR · 344, 347, 360, 369, 378, 380, 384, 410
 DoR effectiveness · 347
 downward spiral · 397
 DTAP · 345, 358, 363, 364, 366, 375, 404, 410
 DTAP environment · 358, 404
 DTAP pipeline · 294
 DTAP street · 345, 364
 DU · 410
 Duration deployment · 364
 DVS · 14, 47, 68, 95, 100, 101, 102, 103, 104, 105, 123, 124, 126, 128, 130, 131, 132, 133, 215, 222, 227, 239, 242, 243, 248, 249, 260, 264, 266, 267, 282, 283, 372, 410

E

E2E · 332, 347, 358, 364, 366, 367, 375, 378, 410
 - acceptance · 364, 378
 - SIT · 347
 - test · 358
 - UAT · 347
 eavesdropping · 169, 171
 eclipse · 170
 e-mail pass around · 345, 397
 emerging design · 94, 229
 enable value stream · 248, 282, 284
 End User eXperience Monitoring · See EUX
 endpoint · 94
 End-to-End · See E2E
 enterprise architect · 278
 enterprise architecture · 241, 244, 245, 266, 267, 297
 Enterprise Resource Planning · See ERP
 Enterprise Service Bus · See ESB
 Entity Relation Diagram · See ERD

environment · 362, 363, 364
 epic · 220, 221, 247, 334, 344, 346, 347, 351, 384, 410
 Epic Solution Approach · See ESA
 epic user story · 221
 equipment failure · 170
 ERD · 410
 ERP · 114, 282, 293, 324, 410
 error path · 397
 ESA · 410
 ESB · 410
 E-shaped · 349, 369, 377, 398
 ETL · 410
 EUX · 410
 event · 344, 345, 352, 365, 366, 367, 377, 402
 - analysis · 366
 - catalogue · 365, 376
 - correlation · 366, 376
 - register · 135
 evidence · 70, 71, 81, 82, 99, 103, 108, 112, 113, 116, 117, 118, 123, 124, 126, 128, 132, 135, 136, 147, 159, 172, 174, 176, 177, 180, 181, 197, 200, 201, 202, 207, 208, 223, 232, 371, 372, 373, 382, 383
 evidence collector · 136, 371
 evidence criteria · 135
 evolutionary project management · 218
 exception · 366
 eXtensible Markup Language · See XML
 external audit · 149, 199, 204, 212, 213, 232
 external auditor · 113
 external issue · 104, 135, 148, 164
 Extract Transform & Load · See ETL
 eXtreme Programming · See XP

F

failure · 394
 failure of communication links · 169
 fast feedback · 321, 334
 FAT · 347, 348, 393, 410
 feature · 221, 247, 346, 347, 357, 397, 398, 407
 feature driven development · 218
 Feature Solution Approach · See FSA
 feature toggle · 397
 feedback · 332, 344, 345, 356, 357, 358, 359, 360, 362, 364, 370, 374, 376, 378, 396, 397, 400, 403, 406
 feedforward · 397
 finding criteria · 135, 174, 176
 fire · 169
 flood · 170
 flow · 3, 343, 344, 346, 361, 366, 369, 396, 399, 400, 401, 402, 403, 404, 406, 407
 four eye-principle · 363
 fragility · 316
 framework · 402

framework of standards · 53, 136
 fraud · 169, 171
 FSA · 410
 Functional Acceptance Test · See FAT
 functionality · 93, 94, 107, 108, 178, 179, 202, 217, 218, 219, 221

G

Game days · 346
 Gaussian distribution · 393, 397
 GCC · 410
 GDPR · 50, 53, 98, 104, 112, 166, 410
 Gene Kim · 397, 402, 406
 General Computer Controls · See GCC
 General Data Protection Regulation · See GDPR
 Generic & Specific Acceptatiecriteria · See GSA
 Gherkin language · 122, 123, 201
 GIT · 249, 410
 Given When Then · 346, 398, See GWT
 Global Information Tracker · See GIT
 goal · 273, 384
 governance · 95, 101, 102, 115, 116, 123, 130, 139, 143, 145, 159, 199, 430
 Graphical User Interface · See GUI
 green build · 334, 345, 361
 green field · 398
 growth model · 112, 122
 GSA · 54, 410
 guarantee · 323
 GUI · 410
 GWT · 42, 60, 354, 398, 410

H

Hand-off Readiness Review · See HRR
 happy path · 358, 393, 394, 398
 hardware · 396, 399, 407
 health model definition · 366
 health model usage · 366
 hide user identity · 169
 holistic approach · 107
 holocracy · 398
 horizontal splitting of feature · 398, 407
 HRM · 29, 93, 118, 119, 121, 261, 262, 263, 322, 324, 328, 330, 339, 340, 410
 HRM officer · 261
 HRM policy · 261
 HRR · 410
 Human Resource Management · See HRM
 hypothesis driven development · 345

I

IaC · 360, 363, 377, 393, 399, 410
 IaC script · 363, 377
 ICT · 399, 410

- ID · 410
- ideal test pyramid · 357, 406
- idempotent · 398
- identification · 360
- IDentifier · See ID
- IDIC · 282
- impact rate · 164
- impact severity · 163, 164, 167, 168
- impact type · 163, 164, 167
- imparative programming · 398
- impediment · 221
- incident criteria · 135
- incidents · 315, 316, 345, 353, 368
- incrementally · 94, 123, 217
- Independent, Negotiable, Valuable, Estimatable, Small and Testable · See INVEST
- information
 - asset · 95, 152, 153, 157, 162, 181, 182
 - management · 102, 227
- information architecture · 245
- Information assets, People, Organisation, Products and services, Systems and processes · See IPOPS
- Information Communication Technology · See ICT
- information radiator · 399
- information security · 17
 - asset · 151, 181
 - auditing engine · 136, 201, 372
 - incident · 139, 140, 142, 160, 165, 174, 178, 199, 209, 212, 213, 373
 - policy · 95, 135, 139, 141, 145, 146, 147, 154, 155, 156, 162, 176, 187, 195, 204, 208
 - risk · 93, 124, 132, 139, 141, 172, 173, 176, 181, 182, 183, 184, 187, 189, 206, 373
- Information Security Management System · See ISMS
- Information Security Value System · See ISVS
- Information Standardisation Organisation · See ISO
- information system · 53
- Information Technology · See IT
- Information Technology Infrastructure Library · See ITIL
- Information Technology Service Management · See ITSM
- Infosec · 399
- Infrastructure as Code · See IaC
- infrastructure component · 399
- infrastructure management · 343, 399
- Infrastructure Risk · See IR
- injection into production · 346
- integrated
 - monitoring · 365
 - pipeline · 361
 - test tooling · 357, 375
 - VSM · 358, 361, 364, 367, 370, 380
- integrated value system · 132
- integrity · 93, 104, 154, 177, 178, 179, 180
- interested parties register · 135, 153
- interested party · 95, 103, 104, 135, 139, 140, 148, 149, 151, 152, 155, 156, 175, 197, 199, 201, 202, 206, 215, 223, 227
- internal audit · 136, 142, 203, 204, 208, 209
- internal audit result · 206
- internal audit time · 206
- internal issue · 104, 161, 162, 164, 166, 184, 186
- internal issue register · 162
- International Standard On Assurance Engagements · See ISAE
- intruder · 154
- intrusion · 94, 128
- INVEST · 398, 410
- IP address · 399
- IPOPS · 135, 159, 160, 161, 162, 163, 164, 410
- IR · 410
- ISAE · 410
- ISAE 3402 · 50, 53
- ISAE 3402 type 2 · 250
- I-shaped · 349, 398
- IshiKawa · 54
- ISMS · 410
- ISO · 410
- ISO 20000 · 50
- ISO 27001 · 50, 53, 72, 85, 95, 97, 98, 100, 103, 113, 118, 128, 129, 132, 136, 146, 148, 151, 152, 157, 162, 166, 176, 184, 191, 193, 194, 199, 201, 204, 205, 207, 208, 211, 226, 232, 249, 250, 293, 302, 382
- ISO 90000 · 50
- ISPL · 282
- issue · 95
- issue register · 135
- IST · 245, 400
- IST – SOLL – Migration path modelling · 97
- IST situation · 385
- ISVS · 95, 97, 100, 101, 102, 103, 104, 105, 109, 122, 123, 124, 126, 128, 129, 130, 131, 132, 133, 135, 136, 138, 139, 140, 141, 142, 143, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 159, 160, 161, 162, 163, 164, 165, 166, 168, 169, 172, 174, 181, 182, 183, 184, 185, 186, 187, 189, 191, 194, 195, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 215, 222, 226, 227, 231, 232, 239, 240, 243, 248, 249, 251, 252, 256, 263, 264, 266, 267, 282, 283, 410
- IT · 397, 401, 406, 410
- iteratively · 94, 123
- ITIL · 95, 100, 101, 102, 131, 228, 241, 248, 249, 282, 410
- ITIL 4 · 2
- ITSM · 401, 410

J

Java Virtual Machine · See JVM
 JIC · 410
 Ji-Kotei-Kanketsu · See, See JKK
 JIT · 399, 400, 410
 JKK · 399, 410
 job description · 146
 Just In Case · See JIC
 Just In Time · See JIT
 JVM · 410

K

Kaizen · 352, 358, 361, 364, 366, 369,
 374, 375, 380, 399, 401
 Kaizen Blitz (or Improvement Blitz) · 400
 Kaizen in advance · 400
 Kanban · 102, 400, 401, 407
 Kanban board · 351
 Key Performance Indicator · See KPI
 kibana dashboard · 400
 knowledge · 346, 366, 368
 knowledge transfer · 325, 368
 KPI · 145, 153, 154, 155, 157, 295, 308,
 309, 344, 352, 355, 358, 359, 361, 364,
 367, 370, 373, 385, 400, 401, 407, 411
 KPI trend measurement · 359, 364, 367,
 370, 378, 381

L

LAN · 411
 late feedback · 384
 latent defect · 400
 Launch Readiness Review · See LRR
 launching guidance · 400
 launching requirement · 345
 laws and regulations · 65, 107, 108, 113,
 147, 166, 229
 LCM · 343, 367, 411
 LDAP · 411
 lead monitor · 97, 126
 lead time · 50
 Lead Time · 400, See LT
 leaked information · 169
 Lean · 406, 407
 Lean indicator · 248, 269, 280, 285, 287,
 288, 297, 304
 Lean software development · 218
 Lean tool · 400
 learning culture · 401
 learning target · 346
 Lemniscaat · 7, 35
 lifecycle · 397, 399
 LifeCycle Management · See LCM
 lightning flash · 169
 Lightweight Directory Access Protocol · See
 LDAP
 limitation · 50, 53, 287, 289

Local Area Network · See LAN
 local binary · 362
 log · 402
 log aggregation · 366
 logging level · 401
 loosely coupled architecture · 401
 loosely coupled services · 401
 loss of electricity · 170
 LRR · 400, 411
 LT · 52, 100, 146, 148, 151, 153, 155,
 160, 164, 165, 167, 168, 173, 181, 184,
 187, 189, 191, 194, 195, 200, 203, 205,
 207, 208, 209, 211, 212, 248, 269, 288,
 289, 361, 400, 411

M

maintenance error · 169
 malicious code · 169
 managed object · 136
 Management Information · See MI
 manual
 - action · 363
 - architecture · 378
 - deployment · 362
 - framework · 366
 - management · 367, 376
 - monitoring · 365, 375
 - plausibility checks · 58
 - provisioning · 352, 365, 366, 367, 381,
 385
 - testing · 356, 371, 374, 382
 manufacturing process · 407
 marker · 42, 139
 MASR · 136, 190, 193, 411
 maturity · 109, 114, 115, 116, 313, 315,
 316, 317, 319, 320, 323, 324, 325, 327,
 329, 330, 333, 334, 335, 336, 339, 343,
 351, 353, 355, 359, 361, 364, 367, 368,
 370, 371, 373, 381, 383
 max duration · 361
 max latency · 366, 375
 Mean Time Between Failure · See MTBF
 Mean Time Between System Incidents ·
 See MTBSI
 Mean Time To Repair · See MTTR
 measurement data · 97
 measurement instruction · 27, 28, 117,
 118
 merging · 360
 meta-data · 385, 394
 metadata · 352, 355, 358, 360, 363, 377,
 380
 methodology · 374
 metric · 370
 MFA · 82, 154, 411
 MI · 411
 Michael Porter · 99, 100, 101
 microservice · 401
 microservice architecture · 401
 Microsoft Operations Framework · See MOF
 mini pipeline · 401

Minimal Viable Metadata · 358
 Minimal Viable Product · See MVP
 Minimum Required Information · See MRI
 mission · 334
 Modify, Avoid, Share, Retain · See MASR
 Module Test · See MT
 MOF · 102, 411
 monitor
 - control · 62
 - facility · 62, 201
 - function · 108, 117
 - layer model · 62
 - matrix · 261
 monitoring · 265, 268, 297, 298, 300, 309, 365, 366, 367, 375, 378, 389, 402, 409
 monitoring archetype · 298
 monitoring tool · 365, 375
 monolithic · 402
 MRI · 401, 411
 MT · 411
 MT member · 149, 150
 MTBF · 411
 MTBSI · 411
 MTTR · 402, 411
 muda · 402
 Multi Factor Authentication · See MFA
 MVP · 240, 246, 247, 346, 352, 385, 411

N

national bank · 149, 150
 NC · 136, 139, 140, 142, 160, 161, 172, 174, 176, 177, 180, 211, 212, 213, 214, 411
 NEN 7510 · 50
 NFR · 102, 395, 402, 411
 Non Conformity · See NC
 Non Functional Requirement · See NFR
 non-compliance · 93

O

OAWOW · 411
 obeya · 402
 obeya room · 344
 object code · 394
 object model · 62
 objectcode · 360
 OLA · 284, 411
 One Agile Way of Working · See OAWOW
 one piece flow · 402
 operational best practice · 95, 159
 Operational Level Agreement · See OLA
 operations · 345, 393, 397, 402, 404, 407
 operations story · 346, 402
 Ops engineer · 1
 Ops liaison · 402
 Ops profile · 346
 organisation archetype · 402
 organisational typology model · 402

outcome · 97, 98, 103, 104, 107, 116, 117, 122, 123, 126, 128, 153, 159, 161, 162, 163, 164, 165, 166, 167, 169, 173, 174, 175, 176, 184, 229, 230
 over-the-shoulder · 403
 ownership · 360

P

PAAS · 364, 411
 package · 403
 pair programming · 357, 378, 395, 403
 password · 85, 170, 177, 178, 179
 PAT · 347, 393, 411
 patch · 343
 pattern · 260, 265, 393, 403
 PBI · 411
 PDCA · 400, 401, 411
 peer review · 357, 378, 403
 peer to peer programming · 395
 People, Process & Technology · See PPT
 PEP · 261, 262, 328
 perfective change · 343
 performance · 244, 252, 269, 273, 283, 287, 288, 289, 290, 297, 305, 344, 358, 361, 364, 366, 370, 395, 401, 407, 411
 Performance · 40, 272
 Performance StressTest · See PST
 PESTLE · 135, 159, 165, 166, 167, 411
 PESTLE classification · 167
 PI · 324
 pipeline · 343, 345, 347, 357, 358, 361, 362, 363, 364, 375, 378, 393, 399, 401, 404, 405, 407
 pipeline phase · 357
 Plan Do Check Act · See PDCA
 Platform As A Service · See PAAS
 Political, Economic, Sociological, Technological, Legislative, Environmental · See PESTLE
 pollution · 170
 POR · 411
 Porter · 247, 248, 249, 274, 281, 299, 300, 301, 390
 post mortem · 403
 power · 24, 26, 95, 113, 114, 258, 323
 PPPT · 251, 263, 280
 PPT · 31, 121, 263, 316, 317, 319, 329, 343, 375, 411
 predictive monitoring · 367
 preventive change · 343
 priority criteria · 135
 privacy authority · 149, 150
 Processing Time · See PT
 product
 - backlog · 23, 105, 113, 115, 215, 218, 219, 221, 222, 226, 227, 229, 232, 240, 246, 251, 252, 253, 257, 258, 259, 264, 265, 268, 279, 289, 290, 301, 305, 325, 331, 334, 344, 347, 351, 372, 382, 384, 403
 - backlog item · 399

- backlog items · 325
- backlog tool · 347
- log file · 366
- owner · 149, 150, 180, 218, 219, 220, 221, 222, 347, 403
- portfolio · 135, 153, 181, 182
- vision · 61, 239, 240, 266, 267, 268, 274, 281, 299, 334, 352, 384

Product Backlog Item · See PBI
 Production AcceptatieTest · See PAT
 production data · 357, 358
 production environment · 102, 122, 142, 201, 228, 260, 268, 298, 384, 401
 production time · 50
 programming paradigm · 403
 Project or Organisational Risk · See POR
 projectteam · 221
 PSQL · 396
 PST · 411
 PT · 52, 146, 148, 151, 153, 155, 160, 164, 165, 167, 168, 173, 181, 184, 187, 189, 192, 194, 195, 200, 203, 205, 207, 208, 209, 211, 212, 248, 269, 288, 361, 411
 pull request process · 403
 Python · 86

Q

QA · 332, 403, 411
 QA department · 116
 QC · 411
 quality · 93, 94, 95, 104, 107, 109, 115, 116, 118, 122, 123, 128, 130, 143, 147, 148, 169, 199, 217, 220, 221, 226, 227, 229, 372
 Quality Assurance · See QA
 Quality Control · See QC
 quality goal · 274

R

RACI · 411
 RASCI · 115, 116, 121, 258, 259, 264, 274, 277, 278, 279, 324, 325, 331, 411
 RASCI set up · 31
 RBAC · 411
 reactive monitoring · 365, 378
 Real User Monitoring · See RUM
 recursion · 100
 redo work · 345
 reduce batch size · 403
 reduce number of handoffs · 403
 refactoring · 109, 229, 315, 360, 378, 380
 regression test · 357
 regulatory obligation · 93
 release · 2, 228, 237, 343, 403

- manager · 403
- pattern · 403
- plan · 334, 352
- planning · 351

- strategy · 363, 378

repository · 357, 359, 360, 361, 363, 366, 375, 377, 380, 394, 395, 403, 404
 REpresentational State Transfer Application Programming Interface · See REST API
 reproduceability · 364
 requirement · 348, 394, 400, 402, 405, 411
 requirement view · 295, 297, 308
 residual risk · 136
 resource · 238
 Responsibility, Accountable, Consulted and Informed · See RACI
 Responsibility, Accountable, Supporting, Consulted and Informed · See RASCI
 REST API · 136, 372, 382
 REST-API · 411
 restrisiko · 196
 retrospective · 384, 397
 Return On Investment · See ROI
 review · 257, 264, 265, 397
 rhythm · 384
 risico · 394, 400
 risk · 344, 358, 361, 364, 367, 380

- analysis · 256, 266, 268, 283, 285, 290
- assessment · 95, 104, 135, 141, 159, 172, 173, 186, 187, 188, 193, 194, 224, 225, 226
- control · 136, 372, 373
- evaluation · 187
- identification · 174, 185, 186, 188
- lifecycle management · 55
- management · 98, 103, 107, 108, 117, 124, 130, 161, 165, 174, 194, 265
- register · 136, 189
- treatment · 136, 141, 193, 224, 225
- treatment option · 105, 141, 159, 173, 189, 190, 191, 192, 193, 194
- treatment plan · 105, 187, 191, 195, 196, 197, 198

 roadmap · 60, 87, 88, 111, 115, 122, 123, 195, 196, 197, 198, 210, 229, 230
 Roadmap · 60, 61, 86
 roadmap planning · 273, 358
 Roadmap to value · 238, 239, 255, 265, 266, 274, 277, 281, 282, 285, 287, 290, 299
 Robotic Process Automation · See RPA
 ROI · 411
 role · 340, 343
 Role-based access control · See RBAC
 root cause · 251, 252, 253, 290
 rootcause analyse · 401
 RPA · 411
 RUM · 411

S

SA · 411
 SABSA goals · 50
 sad path · 358, 403
 SAFe · 411

- SAFe framework · 25, 115
- safety check · 403
- Sarbanes Oxley · See SoX
- SAT · 411
- SBAR · 404, 411
- SBB · 41, 138, 272, 339, 412
- SBB-A · 293, 307, 412
- SBB-I · 290, 293, 307, 412
- SBB-T · 41, 138, 272, 290, 293, 308, 339, 412
- SBB-T diagram · 290
- SBI · 264, 265
- SBL · 223, 225
- Scaled Agile Framework · See SAFe
- S-CI · 358, 411
- scope · 95, 98, 104, 132, 135, 139, 140, 145, 151, 152, 153, 154, 155, 156, 159, 176, 181, 183, 187, 196, 198, 199, 203, 204, 206, 215, 230
- SCOR · 282
- Scrum master · 114, 218, 220, 221
- Scrum team · 220, 221
- secure code review · 358
- Secure Sockets Layer · See SSL
- security · 93, 357, 358, 360, 361, 366, 367, 378, 395, 402, 403, 404, 406
 - analyst · 160, 161, 165, 169, 173, 182, 184, 187, 189, 190, 192, 194, 195, 196, 200, 201, 209, 210, 211, 212, 213
 - control · 124
 - goal · 97, 115, 129, 139, 233
 - manager · 140
 - officer · 140, 141, 142, 395
 - policy · 129
 - practice · 95, 129, 130, 135, 143, 145, 147, 148, 151, 153, 155, 159, 199, 212, 213, 215, 222, 230, 231
- Security Acceptance Test · See SAT
- self service capability · 404
- service · 411
 - desk · 257, 267, 280
 - management process · 343
 - monitoring · 365, 375
 - organisation · 370, 381
 - portfolio · 135, 153, 181, 182
- Service Level Agreement · See SLA
- service level manager · 240, 252, 253, 255, 257, 258, 259, 260, 265, 267, 268, 269, 272, 273, 277, 279, 280, 284
- Service Value System · See SVS
- shared deployment script · 363
- shared goals · 404
- Shareholder · 149, 150
- shift-left organisation · 111
- short lived branch · 326, 327
- silo · 326, 407
- Simian army · 346, 404, 406
- Simple Network Management Protocol · See SNMP
- SIP · 259, 264, 268, 294
- SIT · 345, 348, 412
- Situation, Background, Assessment, Recommendation · See SBAR
- skill · 239, 261, 262, 263, 264
- skills · 398
- SLA · 366, 367, 378, 412
- SLA norm · 113, 139, 264, 265, 277, 279, 295, 298, 366, 385
- slow feedback · 18, 108
- SM · 412
- SMART · 135, 153, 154, 401, 412
- SMART goal · 266
- SME · 324, 412
- smoke testing · 404
- SNMP · 412
- SoA · 136, 412
- social engineering · 170
- SoE · 93, 94, 95, 114, 324, 405, 412
- software · 394, 405, 407
- Software Configuration Item · See S-CI
- software package · 282, 293
- software work · 349
- SoI · 93, 94, 95, 405, 412
- SOLL · 245, 400
- SOLL situation · 385
- SoR · 93, 94, 95, 114, 324, 405, 412
- sourcecode · 357, 359, 360, 361, 375, 377, 379, 380, 381, 394, 395, 397, 404, 405, 406
- SoX · 50, 412
- Specific, Measurable, Accountable, Realistic, Timely · See SMART
- Spotify · 115
- Spotify model · 324
- sprint · 95, 219, 222, 225, 226, 247, 384, 385, 397
 - backlog · 221, 255
 - backlog tool · 347
 - execution · 397
 - goal · 344
 - length · 257
 - planning · 89, 244, 246, 253, 257, 265, 266, 273, 351, 384, 397
 - retrospective · 315
- SQL · 412
- squad · 357, 358, 359
- SRC · 135, 136, 145, 146, 147, 149, 150, 151, 152, 153, 154, 155, 156, 180, 187, 196, 198, 203, 204, 208, 209, 210, 212, 213, 250
- SRC-board · 136, 146, 147, 150, 196
- SRC-board member · 149
- SRG · 358, 360, 363, 366, 375, 378, 380, 412
- SSL · 412
- ST · 348, 412
- stakeholder · 93, 99, 104, 107, 121, 139, 140, 145, 148, 149, 156, 174, 220, 221, 327, 340, 343, 355, 398
- standard deviation · 404
- standard operations · 404
- Standard Rules & Guidelines · See SRG
- stand-up · 397
- Statement of Applicability · See SoA

static analysis · 405
 story · 247, 347, 358, 360
 Strategic IS Architecture · See SA
 Strategic Match · See SM
 strategy · 313, 334, 354, 358, 359, 360, 362, 363, 365, 368, 370, 378
 Strength, Weakness, Opportunities, Threats · See SWOT
 strike · 170
 Structured Query Language · See SQL
 Subject Matter Expert · See SME
 supplier · 149
 sustainable · 147, 166, 349
 SVS · 14, 47, 53, 68, 95, 97, 100, 101, 102, 103, 104, 105, 123, 124, 126, 128, 130, 131, 132, 133, 159, 160, 161, 186, 196, 215, 227, 239, 240, 241, 242, 243, 248, 249, 251, 252, 256, 260, 263, 264, 266, 267, 282, 283, 412
 SWOT · 273, 274, 283, 293, 301, 302, 303, 412
 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 context diagram · 135, 153
 system development · 93, 96, 100, 215, 216, 217, 218, 228
 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

taak · 393
 tag · 379, 380
 target · 97, 98, 107, 108, 109, 112, 116, 123, 127, 139, 155, 157
 Target Operating Model · See TOM
 task · 247, 346, 399
 TCO · 412
 TCP · 412
 TDD · 316, 345, 348, 354, 356, 374, 405, 412
 Team Foundation Server · See TFS
 technical debt · 111, 115, 230, 315, 316, 321, 325, 336, 341, 360, 361, 363, 370, 380, 396, 397
 technical debt backlog · 360, 361, 363, 376
 technical excellent · 349
 Technical Information Security Officer · See TISO
 Technical Uncertainty · See TU
 technology adaption curve · 405
 technology executive · 405
 telemetry · 345, 346
 template · 138, 147, 149, 166, 167, 174, 175, 176, 177, 178, 180, 181, 182, 184, 185, 186, 187, 188, 190, 191, 192, 195, 197, 201, 205, 206, 211, 212, 213, 225, 226, 232
 terrorist attack · 170
 test
 - automation · 357
 - case · 345, 347, 356, 357, 358, 377, 380, 393, 394, 395
 - criteria · 173, 174, 186
 - data · 357, 358
 - data generating tool · 357
 - generation · 357
 - harness · 405
 - lifecycle · 357
 - management · 347, 357, 358, 376
 - object · 357, 358
 - pattern · 356, 357
 - script · 357, 377
 - strategy · 356, 357, 358, 378
 - tool · 356
 - type · 357, 358
 Test Driven Development · See TDD
 tester · 397
 TFS · 412
 The Agile Manifesto · 405
 the ideal testing automation pyramid · 406
 The Lean movement · 406
 the non-ideal testing automation inverted pyramid · 406
 The Three Ways · 402, 406
 theft · 169
 theme · 246, 346, 347, 384
 theory of constraints · 406
 threat · 169, 171
 Time To Market · See TTM
 time traveling · 364, 378
 TISO · 181, 412
 TOM · 42, 45, 97, 98, 101, 117, 123, 124, 132, 412
 tool assisted code rev · 345
 tool-assisted code review · 406
 Total Cost of Ownership · See TCO
 Toyota Kata · 407
 Toyota Production System · See TPS
 TPS · 99, 247, 391, 412
 traceability · 18, 108, 112, 113, 210, 233, 330, 334, 335, 344, 347, 360, 361, 363, 375, 378, 381
 transformation team · 407
 Transmission Control Protocol · See TCP
 transport · 154
 trend analysis · 361
 tribe · 357, 358
 trigger criteria · 174
 trunk · 404
 trust · 349
 T-shaped · 349, 398
 TSQL · 396
 TTM · 412
 TU · 412

U

UAT · 364, 412
 UML · 412
 Unified Modeling Language · See UML
 uniform
 - meta data · 358
 - test process · 357
 - test terminology · 357, 376
 - test tooling · 357, 375
 - testproces · 376
 - WoW · 322, 323
 Unit Test · See UT
 unit test case · 42, 357
 upfront design · 94
 use case · 95, 108, 135, 136, 137, 138, 140, 141, 142, 146, 147, 148, 151, 153, 154, 155, 156, 157, 160, 161, 164, 165, 168, 169, 172, 174, 175, 176, 181, 182, 183, 184, 187, 189, 190, 191, 194, 195, 199, 201, 203, 204, 205, 206, 207, 208, 209, 211, 212, 213, 224, 231, 238, 248, 269, 271, 272, 273, 285, 299, 302, 304
 Use Case · 40, 41, 43, 272, 314, 337, 354
 use case diagram · 40, 95, 135, 136, 137, 138, 224, 231
 User Acceptance Test · See UAT
 user error · 169
 User eXperience Design · See UX design
 user interface · 86
 user story · 93, 94, 221, 346
 UT · 358, 412
 UX design · 412

V

valuable software · 349
 value
 - chain · 45, 95, 99, 100, 101, 102, 107, 116, 121, 126, 128, 129, 130, 135, 138, 139, 143, 145, 151, 152, 153, 164, 167, 230, 334
 - stream · 39, 45, 48, 52, 93, 95, 97, 98, 99, 100, 101, 102, 103, 108, 109, 112, 116, 122, 127, 128, 129, 131, 135, 136, 138, 139, 140, 143, 152, 153, 161, 164, 167, 174, 175, 176, 177, 178, 179, 214, 222, 228, 229, 230, 271, 287, 288, 299, 348, 358, 360, 361, 364, 367, 369, 370, 372, 373, 374, 375, 376, 380, 397, 401, 404, 405, 407, 412
 - system · 101, 102, 103, 123, 126, 127, 128, 132, 133, 373
 value stream
 - canvas · 275, 280, 287, 288, 289, 304, 305, 334, 354
 - goal · 240, 251, 252, 253, 256, 257, 258, 264, 266, 268, 274, 275, 277,

278, 279, 280, 283, 284, 288, 293, 299, 301, 303, 304, 385
 - manager · 140, 241, 258, 263, 269, 273, 275, 279, 289
 - mapping · 360, 361, 374
 - mapping model · 268, 285
 - owner · 274, 278, 279
 Value Stream Mapping · See VSM
 vandalism · 170
 velocity · 222, 344, 394
 velocity trend · 344
 version · 343
 version control · 360, 375
 versioning · 356, 359, 360
 vertical splitting of feature · 407
 violation of law · 170
 virtualised environment · 407
 vision · 95, 112, 238, 256, 313, 321, 322, 323, 334, 343, 347, 352, 361
 visualisation · 322, 323, 407
 Voice over Internet Protocol · See VOIP
 VOIP · 412
 VSM · 42, 294, 361, 407, 412
 vulnerability scan · 267, 295
 vulnerable · 169, 170, 175

W

walking skeleton · 407
 WAN · 412
 war room · 402
 waste · 246, 248, 285, 288, 289, 297, 346, 361, 364, 377, 394, 396, 399, 400, 402, 406, 407
 waste record · 361, 377
 waste reductie · 407
 waterfall project · 93
 Way of Working · See WoW
 Westrum · 402, 403
 Wide Area Network · See WAN
 Windows Management Instrumentation · See WMI
 WIP · 412
 WMI · 412
 Work In Progress · See WIP
 work item · 247
 workflow · 396
 WoW · 321, 324, 325, 412

X

XML · 412
 XP · 102, 218, 412

Z

Zachman · 245, 297

Epilogue

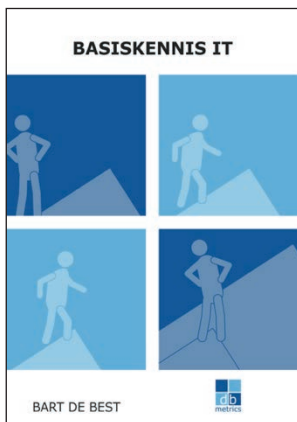
My experience is that the ideas I capture in an article or book continue to evolve. If you are going to work with a certain topic from this book in your own DevOps organisation, I advise you to contact me. Perhaps there are additional articles or experiences in this area that I can share with you. This 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 managers, advising management organisations in directing the management organisation, 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

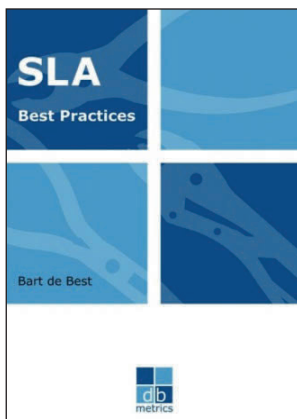
De eerste stap van een leven lang leren.

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 heraanboren, 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



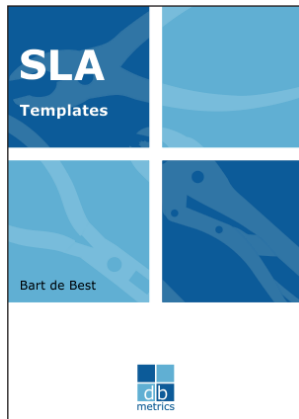
Cloud SLA

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

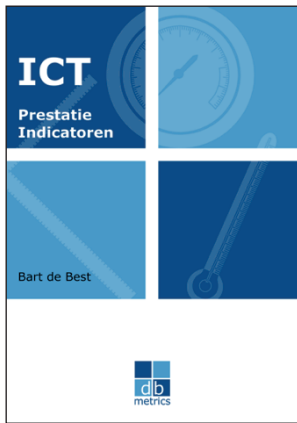
A complete set of SLA templates

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)
- Document Financial Agreements (DFA)
- Service Catalogue
- External Spec Sheet (ESS)
- Internal Spec Sheet (ISS)
- Service Quality Plan (SQP)
- Service Improvement Program (SQP)

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



ICT Prestatie-indicatoren

De beheerorganisatie meetbaar gemaakt.

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

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

Auteur : Bart de Best
 Uitgever : Leonon Media, 2011
 ISBN (NL) : 978 90 7150 1470



Quality Control & Assurance

Kwaliteit op maat.

De business stelt steeds hogere eisen aan de ICT-services die ICT-organisaties leveren. Niet alleen nemen de eisen van de overheid toe in de vorm van wet- en regelgeving, ook de dynamiek van de markt wordt hoger en de levenscyclus van business producten korter. De reactie van veel ICT-organisaties hierop is het hanteren van kwaliteitsmodellen zoals COBIT, ITIL, TOGAF en dergelijke. Helaas verzandt het toepassen van de best practices van deze modellen vaak omdat het model als doel wordt verklaard, hierdoor ontstaat veel overhead. Nut en noodzaak worden niet onderscheiden. In het beste geval is de borging van kwaliteit een golfbeweging met pieken en dalen waarop maar weinig grip op te

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

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



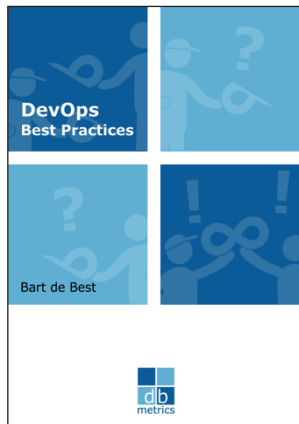
Agile Service Management with Scrum in Practice

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



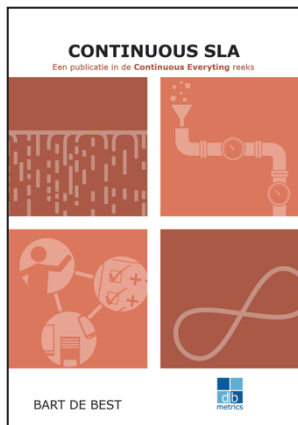
Continuous Planning

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



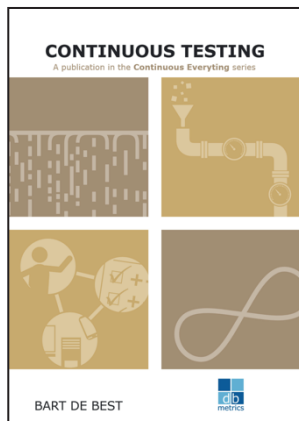
Continuous Design

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



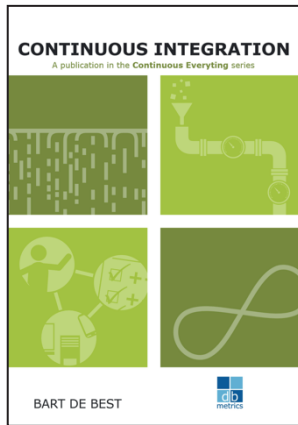
Continuous Testing

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 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 be 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

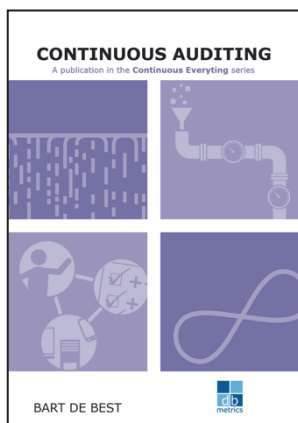
A publication in the Continuous Everything series.

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 realizing, putting into production, and managing the new or modified products and services at a rapid pace.

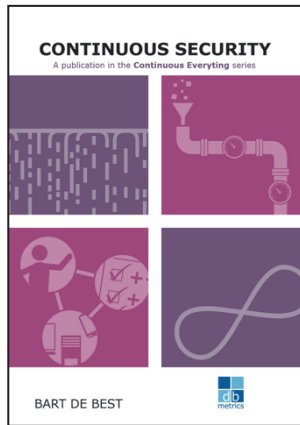
As a result, compliance risks are prevented by already thinking about which risks to mitigate or eliminate from the requirements and the design based on them.

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, realizing 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



Continuous Operations

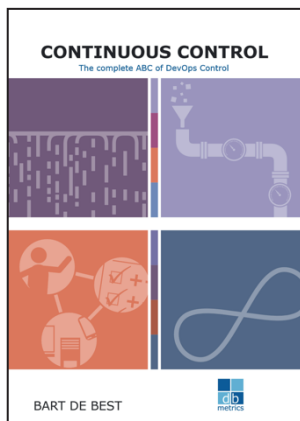
A publication in the Continuous Everything series.

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 Audit, Continuous Security, Continuous SLA 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 91480 195
ISBN (UK)	: 978 94 91480 201



Continuous Everything

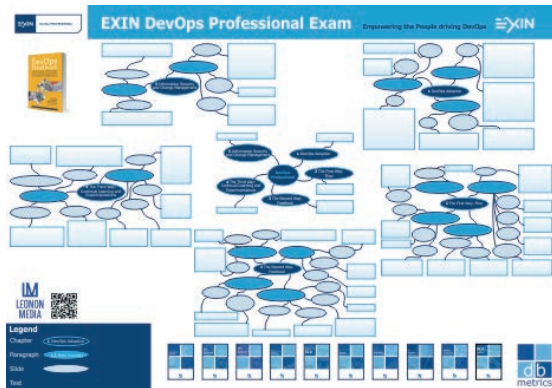
A publication in the Continuous Everything series.

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



DevOps Poster

DevOps Professional Exam Poster

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

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

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

CONTINUOUS CONTROL

The complete ABC of DevOps Control

Bart de Best



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 4 Continuous Everything books, namely:

1. Continuous Auditing
2. Continuous Security
3. Continuous SLA
4. Continuous Assessment

For each Continuous Everything aspect area it is indicated how to organize it in your organization 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.

ISBN 978-9-491480-20-1



9 789491 480201 >