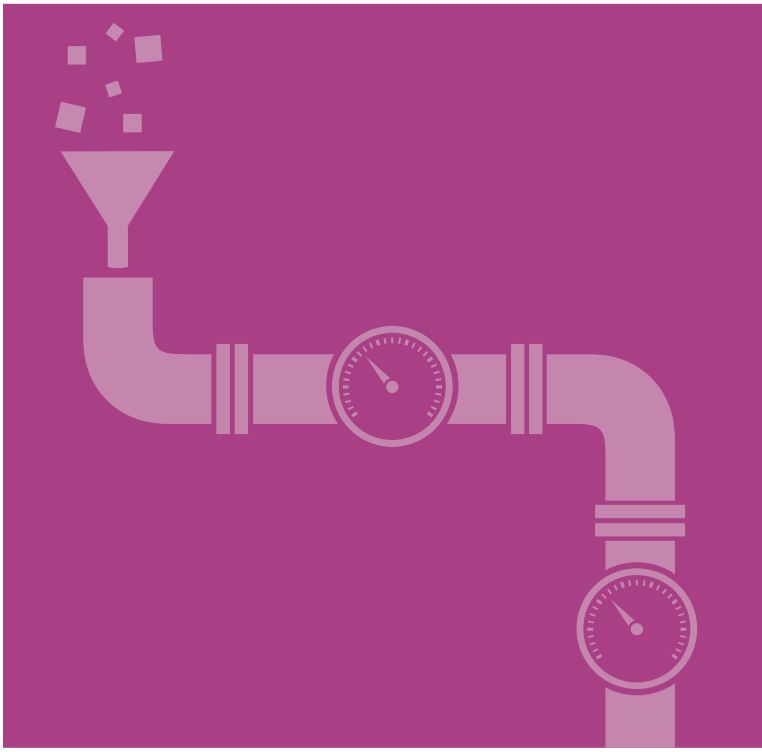
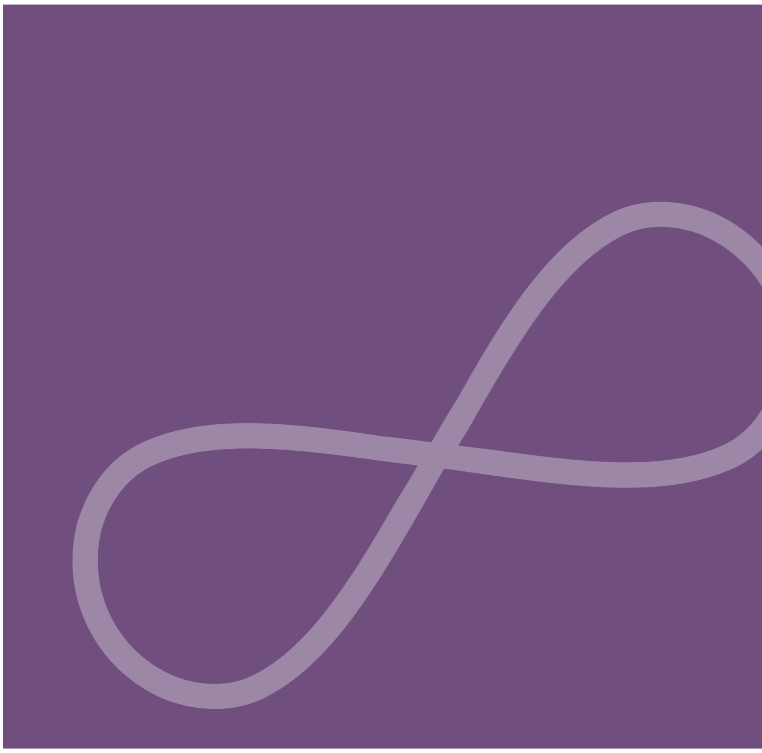
CONTINUOUS SECURITY

A publication in the **Continuous Everyting** series





BART DE BEST

db metrics

DevOps
Continuous Security
Best Practices

A publication in the Continuous Everything series

Bart de Best

Edited by
Louis van Hemmen

Colophon

More information about this and other publications can be obtained from:

Leonon Media

(0)572 - 851 104

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

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

Title : DevOps Continuous Security

Subtitle : A publication in the Continuous Everything series

Date : 15 December 2022

Author : Bart de Best

Publisher : Leonon Media

ISBN13 : 978 94 91480 188

Edition : First press, third edition, 15 December 2022

"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 GOAL 1
1.2 TARGET AUDIENCE 1
1.3 BACKGROUND 1
1.4 STRUCTURE 4
1.4.1 CHAPTER 2: BASIC CONCEPTS AND BASIC TERMS 4
1.4.2 CHAPTER 3: CONTINUOUS SECURITY DEFINITION 4
1.4.3 CHAPTER 4: CONTINUOUS SECURITY ANCHORAGE 4
1.4.4 CHAPTER 5: CONTINUOUS SECURITY ARCHITECTURE 4
1.4.5 CHAPTER 6: CONTINUOUS SECURITY DESIGN 4
1.4.6 CHAPTER 7: CONTINUOUS SECURITY BEST PRACTICES 4
1.4.7 CHAPTER 8: GOVERNANCE SECURITY PRACTICES 4
1.4.8 CHAPTER 9: RISK SECURITY PRACTICES 4
1.4.9 CHAPTER 10: QUALITY SECURITY PRACTICES 4
1.4.10 CHAPTER 11: CONTINUOUS SECURITY VERSUS AGILE SCRUM 4
1.4.11 CHAPTER 12: CONTINUOUS SECURITY VERSUS DEVOPS 5
1.4.12 CHAPTER 13: CONTINUOUS SECURITY ASSESSMENT 5
1.5 APPENDICES 5
1.6 READING GUIDELINES 5
2 BASIC CONCEPTS AND BASIC TERMS 7
2.1 BASIC CONCEPTS 7
2.1.1 CONTINUOUS CONTROL 7
2.1.2 CONTINUOUS SECURITY PYRAMID 7
2.1.3 VALUE CHAIN 9
2.1.4 VALUE STREAM 9
2.1.5 DVS, SVS AND ISVS 10
2.1.6 POSITIONING CONTINUOUS SECURITY 11
2.1.7 THE THREE PERSPECTIVES OF INFORMATION SECURITY 12
2.2 BASIC TERMS 13
2.2.1 RISK MANAGEMENT TERMS 13
2.2.2 VALUE SYSTEM TERMS 15
3 CONTINUOUS SECURITY DEFINITION 17
3.1 BACKGROUND 17
3.2 DEFINITION 17
3.3 APPLICATION 17
3.3.1 PROBLEMS TO BE SOLVED 17
3.3.2 THE ROOT CAUSE 18
4 CONTINUOUS SECURITY ANCHORAGE 21
4.1 THE CHANGE PARADIGM 21
4.2 VISION 22
4.2.1 WHAT DO WE WANT? 22
4.2.2 WHAT DO WE NOT WANT? 23
4.3 POWER 23
4.3.1 WHAT DO WE WANT? 24
4.3.2 WHAT DO WE NOT WANT? 26
4.4 ORGANISATION 27
4.4.1 WHAT DO WE WANT? 27
4.4.2 WHAT DO WE NOT WANT? 27

4.5 Resources 28
4.5.1 What do we want? 28
4.5.2 What do we not want? 29
5 CONTINUOUS SECURITY ARCHITECTURE 31
5.1 Architecture principles 31
5.1.1 General 31
5.1.2 People 31
5.1.3 Process 31
5.1.4 Technology 34
5.2 Architecture models 34
5.2.1 Continuous security pyramid model 34
5.2.2 Control Model 36
5.2.3 Quality Control & Assurance Model 37
5.2.4 Value Systems 38
5.2.5 Information Security Value Systems 38
5.2.6 Service Value Systems 41
5.2.7 Development Value Systems 41
5.2.8 Integrated value systems 42
6 CONTINUOUS SECURITY DESIGN 45
6.1 Continuous security value stream 45
6.2 Continuous security use case diagram 46
6.3 Continuous security use case 47
7 CONTINUOUS SECURITY BEST PRACTICES 53
7.1 Best practices 53
7.2 Value stream examples 53
8 GOVERNANCE SECURITY PRACTICES 55
8.1 Scope governance security practices 55
8.2 Get Top Management Commitment 55
8.2.1 Use case 56
8.2.2 Definition 57
8.2.3 Goal 57
8.2.4 Example 57
8.2.5 Best practices 58
8.3 Determine interested parties 58
8.3.1 Use case 58
8.3.2 Definition 59
8.3.3 Goal 59
8.3.4 Example 59
8.3.5 Best practices 59
8.4 Determine Scope 61
8.4.1 Use case 61
8.4.2 Definition 62
8.4.3 Goal 62
8.4.4 Example 62
8.5 Determine goals 63
8.5.1 Use case 63
8.5.2 Definition 64
8.5.3 Goal 64
8.5.4 Example 64

8.5.5 Best practices 65
8.6 Determine information security policies 65
8.6.1 Use case 65
8.6.2 Definition 66
8.6.3 Goal 66
8.6.4 Example information security policy 66
8.6.5 Example Code of Conduct 67
8.6.6 Best practices Goals 67
8.6.7 Best practices Code of Conduct 68
9 RISK SECURITY PRACTICES 69
9.1 Scope risk security practices 69
9.2 Determine Issues - Internal 70
9.2.1 Use case internal issues 70
9.2.2 Definition 71
9.2.3 Goal 71
9.2.4 Example 72
9.2.5 Best practices – WoW IPOPS 72
9.3 Determine issues - external 74
9.3.1 Use case external issues 74
9.3.2 Definition 75
9.3.3 Goal 75
9.3.4 Example 75
9.3.5 Best practices – WoW PESTLE 76
9.4 Determine CRAMM issues 78
9.4.1 Use case CRAMM issues 78
9.4.2 Definition 79
9.4.3 Goal 79
9.4.4 Example 79
9.4.5 Best practices – WoW CRAMM 80
9.5 Determine Risk criteria 82
9.5.1 Use case risk criteria 82
9.5.2 Definition 84
9.5.3 Goal 84
9.5.4 Example 84
9.6 Determine Information Assets 91
9.6.1 Use case information assets 91
9.6.2 Definition 92
9.6.3 Goal 92
9.6.4 Example 92
9.7 Identify risks 93
9.7.1 Definition 94
9.7.2 Goal 94
9.7.3 Example 95
9.7.4 Best practices – WoW Risk identification 95
9.8 Perform Risk assessment 96
9.8.1 Definition 97
9.8.2 Goal 97
9.8.3 Example 97
9.8.4 Best practices – WoW Risk assessment 98
9.9 Perform Risk Treatment - options 99
9.9.1 Definition 100

9.9.2 Goal 100
9.9.3 Example 100
9.9.4 Best practices – WoW Risk Treatment 100
9.10 Determine Risk Treatment - controls 101
9.10.1 Definition 102
9.10.2 Goal 102
9.10.3 Example 102
9.10.4 Best practices – WoW Risk Treatment 103
9.11 Perform Risk Treatment – existing controls 104
9.11.1 Definition 105
9.11.2 Goal 105
9.11.3 Example 105
9.12 Realise Controls 105
9.12.1 Definition 107
9.12.2 Goal 107
9.12.3 Example 107
9.12.4 Best practices – WoW treatment plan 107
10 QUALITY SECURITY PRACTICES 109
10.1 Scope quality security practices 109
10.2 Monitor effectiveness controls 109
10.2.1 Definition 111
10.2.2 Goal 111
10.2.3 Example 111
10.2.4 Best practices – WoW monitor facility 111
10.3 Perform internal audit - Plan 113
10.3.1 Definition 114
10.3.2 Goal 114
10.3.3 Example 114
10.3.4 Best practices – WoW internal audit 114
10.4 Perform internal audit - Criteria 115
10.4.1 Definition 116
10.4.2 Goal 116
10.4.3 Example 116
10.5 Perform internal audit - Performance 116
10.5.1 Definition 118
10.5.2 Goal 118
10.5.3 Example 118
10.6 Perform internal audit - Report 118
10.6.1 Definition 119
10.6.2 Goal 119
10.6.3 Example 119
10.7 Improve Continuous - incidents 119
10.7.1 Definition 121
10.7.2 Goal 121
10.7.3 Example 121
10.8 Improve Continuous – non-conformities 121
10.8.1 Definition 122
10.8.2 Goal 122
10.8.3 Example 122
10.9 Improve Continuous – CSI 122
10.9.1 Definition 123

10.9.2 GOAL 123
10.9.3 EXAMPLE 123

11 CONTINUOUS SECURITY VERSUS AGILE SCRUM 125
11.1 POSITIONING 125
11.2 AGILE MANIFESTO 125
11.3 AGILE METHODS 128
11.4 AGILE SCRUM 128
11.4.1 THE AGILE SCRUM APPROACH 129
11.4.2 THE AGILE SCRUM DEVELOPMENT PROCESS 129
11.4.3 THE AGILE SCRUM TERMS 130
11.5 CONTINUOUS SECURITY IN AGILE SCRUM 132
11.6 THE DIFFERENCE 136

12 CONTINUOUS SECURITY VERSUS DEVOPS 137
12.1 DEVOPS POSITIONING 137
12.2 DEVOPS CONCEPT 137
12.2.1 DEVOPS – THE ORIGIN 137
12.2.2 DEVOPS – WHAT IS IT? 138
12.2.3 DEVOPS VISUALISED 138
12.2.4 CONTINUOUS PLANNING 139
12.2.5 CONTINUOUS DESIGN 139
12.2.6 CONTINUOUS TESTING 139
12.2.7 CONTINUOUS INTEGRATION 139
12.2.8 CONTINUOUS DEPLOYMENT 139
12.2.9 CONTINUOUS MONITORING 140
12.2.10 CONTINUOUS LEARNING 140
12.2.11 CONTINUOUS ASSESSMENT 140
12.3 CONTINUOUS SECURITY IN DEVOPS 140
12.4 THE DIFFERENCE 142

13 CONTINUOUS SECURITY ASSESSMENT 145
13.1 WHAT IS THE CE-MODEL 145
13.2 MATURITY DIMENSIONS 148
13.3 DEVOPS CE MODEL, CY 148

APPENDIX A, LITERATURE LIST 155
APPENDIX B, GLOSSARY 159
APPENDIX C, ABBREVIATIONS 175
APPENDIX D, WEBSITES 179
APPENDIX E, INDEX 181

Figures

Figure 1-1, DevOps Lemniscate. 1
Figure 1-2, SoR, SoE en SoI (source HSO the result company). 3
Figure 2-1, Continuous control. 7
Figure 2-2, Continuous security pyramid. 8
Figure 2-3, Value chain of Porter, bron: [boek Michael Porter]. 9
Figure 2-4, Recursieve Value chain of Porter, bron: [Michael Porter 1998]. 10
Figure 2-5, Recursieve Value chain of Porter, bron: [Michael Porter] 11
Figure 2-6, The composition a value system. 11
Figure 2-7, The structure of a business value chain. 12
Figure 2-8, The three perspectives of information security. 13
Figure 2-9, Risk terms. 14
Figure 2-10, Value system terms. 15
Figure 4-1, Change paradigm. 21
Figure 4-2, The change paradigm - Vision. 22
Figure 4-3, The change paradigm - power. 24
Figure 4-4, The change paradigm - organisation. 27
Figure 4-5, The change paradigm - resources. 28
Figure 5-1, Continuous security pyramid. 35
Figure 5-2, Continuous security pyramid depicted on the DevOps Lemniscate. 35
Figure 5-3, Continuous security pyramid with deliverables and questions to be answered. 36
Figure 5-4, Continuous security pyramid model mapped to continuous control model. 37
Figure 5-5, Quality Control & Assurance model. 37
Figure 5-6, Recursive value chain. 38
Figure 5-7, Information security value chain. 39
Figure 5-8, Information Security Value System. 39
Figure 5-9, Information security practices. 40
Figure 5-10, Information Security Value System overview. 41
Figure 5-11, Service value chain. 41
Figure 5-12, Development value chain. 42
Figure 5-13, Continuous security pyramid depicted on the ISVS, DVS and SVS models. 42
Figure 5-14, Information security perspectives. 43
Figure 6-1, Continuous security value stream. 45
Figure 6-2, Use case diagram for continuous security. 47
Figure 7-1, Information security practices. 53
Figure 7-2, Information Security Value Streams. 54
Figure 8-1, Governance security practices. 55
Figure 9-1, Risk security practices. 69
Figure 9-2, CRAMM model. 81
Figure 9-3, Asset register. 93
Figure 9-4, Risk lifecycle. 95
Figure 10-1, Quality Security practices. 109
Figure 10-2, Monitor architecture for monitoring the effectiveness of controls. 112
Figure 11-1, Positioning Agile and Agile Scrum. 125
Figure 11-2, Agile Scrum development process. 129
Figure 11-3, Agile Scrum teams. 130
Figure 11-4, Continuous security use case diagram. 134
Figure 12-1, Positioning DevOps. 137
Figure 12-2, Continuous security depicted on the DevOps Lemniscate. 138
Figure 12-3, Continuous security use case diagram. 141
Figure 13-1, DevOps CE-Spider model. 147

FIGURE 13-2, DEVOPS CY-SPIDER MODEL. 151

Tables

TABLE 1-1, CONTINUOUS EVERYTHING ASPECTS. 2
TABLE 1-2, APPENDICES. 5
TABLE 3-1, COMMON PROBLEMS WHEN USING CONTINUOUS SECURITY. 18
TABLE 6-1, TERMS PER ISVS USE CASE. 46
TABLE 6-2, USE CASE TEMPLATE. 48
TABLE 6-3, USE CASE FOR CONTINUOUS SECURITY. 52
TABLE 8-1, USE CASE 'GET TOP MANAGEMENT COMMITMENT'. 57
TABLE 8-2, EXAMPLE OF A STATEMENT OF COMMITMENT. 57
TABLE 8-3, DETERMINE INTERESTED PARTY. 59
TABLE 8-4, INTERESTED PARTIES - INFLUENCE AND INTEREST. 60
TABLE 8-5, INTERESTED PARTIES – REGISTER. 60
TABLE 8-6, INTERESTED PARTIES – REGISTER EXPLANATION 61
TABLE 8-7, SCOPE - USE CASE. 62
TABLE 8-8, SCOPE - DEFINITION. 63
TABLE 8-9, SCOPE - USE CASE. 64
TABLE 8-10, GOALS. 65
TABLE 8-11, INFORMATION SECURITY POLICY - USE CASE. 66
TABLE 8-12, CODE OF CONDUCT EXAMPLE FOR BUSINESS AND IT. 67
TABLE 8-13, CODE OF CONDUCT EXAMPLE FOR IT. 67
TABLE 9-1, USE CASE 'BEPAAL INTERNAL ISSUES'. 71
TABLE 9-2, EXAMPLES OF INTERNAL ISSUES FACTORS. 72
TABLE 9-3, EXAMPLES VAN IPOPS FACTORS. 73
TABLE 9-4, EXAMPLE IPOPS CLASSIFICATION TEMPLATE. 73
TABLE 9-5, USE CASE 'DETERMINE EXTERNAL ISSUE'. 75
TABLE 9-6, EXAMPLES OF EXTERNAL ISSUES. 76
TABLE 9-7, EXAMPLES VAN PESTLE FACTORS. 76
TABLE 9-8, EXAMPLE PESTLE CLASSIFICATION TEMPLATE. 77
TABLE 9-9, USE CASE 'DETERMINE CRAMM ISSUE'. 79
TABLE 9-10, EXAMPLES OF CRAMM THREATS. 80
TABLE 9-11, TEMPLATE CRAMM-ANALYSE. 81
TABLE 9-12, USE CASE 'DETERMINE RISK CRITERIA'. 84
TABLE 9-13, RISK PRIORITY CRITERIA. 85
TABLE 9-14, RISK PRIORITY CRITERIA. 86
TABLE 9-15, RISK PRIORITY CRITERIA. 86
TABLE 9-16, INTERNAL AUDIT FINDING CRITERIA. 87
TABLE 9-17, THE IMPACT CODE OF INFORMATION SECURITY INCIDENTS. 88
TABLE 9-18, THE IMPACT CODE OF INFORMATION SECURITY INCIDENTS. 90
TABLE 9-19, INFORMATION SECURITY INCIDENT PRIORITY TABLE. 90
TABLE 9-20, INFORMATION SECURITY INCIDENT EVIDENCE MATRIX. 91
TABLE 9-21, USE CASE 'DETERMINE INFORMATION SECURITY ASSETS'. 92
TABLE 9-22, USE CASE 'IDENTIFY RISKS'. 94
TABLE 9-23, IDENTIFIED RISK. 95
TABLE 9-24, TEMPLATE RISK IDENTIFICATION. 96
TABLE 9-25, EXPLANATION OF RISK IDENTIFICATION ITEMS. 96
TABLE 9-26, USE CASE 'IDENTIFY RISKS'. 97
TABLE 9-27, TEMPLATE RISK IDENTIFICATION. 98
TABLE 9-28, TEMPLATE RISK ASSESSMENT. 98

Table 9-29, Template Risk assessment.....98
Table 9-30, Use case 'Risk treatment options risks'.....100
Table 9-31, MASR treatment options.....100
Table 9-32, Explanation treatment options.....101
Table 9-33, Use case 'Assigning risk controls to risks'.....102
Table 9-34, MASR treatment options.....103
Table 9-35, Use case 'Assigning risk controls to risks'.....105
Table 9-36, Use case 'Drawing up a risk treatment plan for a control'.....106
Table 9-37, Treatment plan template.....107
Table 10-1, Use case 'Monitor effectiveness controls'.....111
Table 10-2, Use case 'Internal audit planning'.....114
Table 10-3, Template internal audit.....115
Table 10-4, Use case 'Internal audit criteria'.....116
Table 10-5, Use case 'Performance internal audit'.....118
Table 10-6, Use case 'Report internal audit'.....119
Table 10-7, Use case 'Information security incidents'.....121
Table 10-8, Use case 'Information security NC'.....122
Table 10-9, Use case 'Information security NC'.....123
Table 10-10, CSI register Example.....124
Table 11-1, Effectiveness aspects of Agile system development.....127
Table 11-2, Efficiency aspects of Agile system development.....127
Table 11-3, Continuous security depicted on Agile Scrum.....133
Table 11-4, Continuous security mapped to Agile Scrum artifacts.....135
Table 11-5, Continuous security mapped to Agile Scrum events.....136
Table 12-1 Tabel 12-1, Continuous everything aspects.....138
Table 12-2, Continuous security depicted on DevOps.....142
Table 13-1, DevOps CE-model.....145
Table 13-2, Continuous everything.....146
Table 13-3, CMMI levels for Continuous everything.....147
Table 13-4, PR-ORG-009. Maturity Levels.....148
Table 13-5, CY maturity characteristics.....150

Appendices

Appendix A, Literature list.....155
Appendix B, Glossary.....159
Appendix C, Abbreviations.....175
Appendix D, Websites.....179
Appendix E, Index.....181

Preface

This book has been compiled based on my experiences in implementing information security in a DevOps context. It's a snapshot of the best practices which I am using now. Given the speed with which the world of DevOps is developing and the need to give you as many images as possible with as little text as possible about using continuous security, I have decided to keep this book Agile. This means that it describes very briefly what important insights I have gained during my role as a consultant, trainer, coach, and examiner with regard to continuous security related work. Where appropriate, I refer to sources that I 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 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.

1 Introduction

Reading guide:
This chapter describes the background of this book (1.1), the intended target group (1.2), the structure (1.3) and finally some tips for handling this book (1.4) and the reading guide (1.5).

1.1 Goal

The goal of this book is to provide basic knowledge regarding continuous security and tips and tricks for applying this aspect area of continuous everything.

1.2 Target audience

The target audience of this book are all involved functions in the DevOps teams. This includes auditors, quality employees, 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, etc. of the information provision through a DevOps method.

1.3 Background

This book contains various techniques to continuously implement compliance so that this quality of the information system grows during production. Continuous security is an integral part of the DevOps Lemniscate as shown in Figure 1-1 and in all steps that are followed.

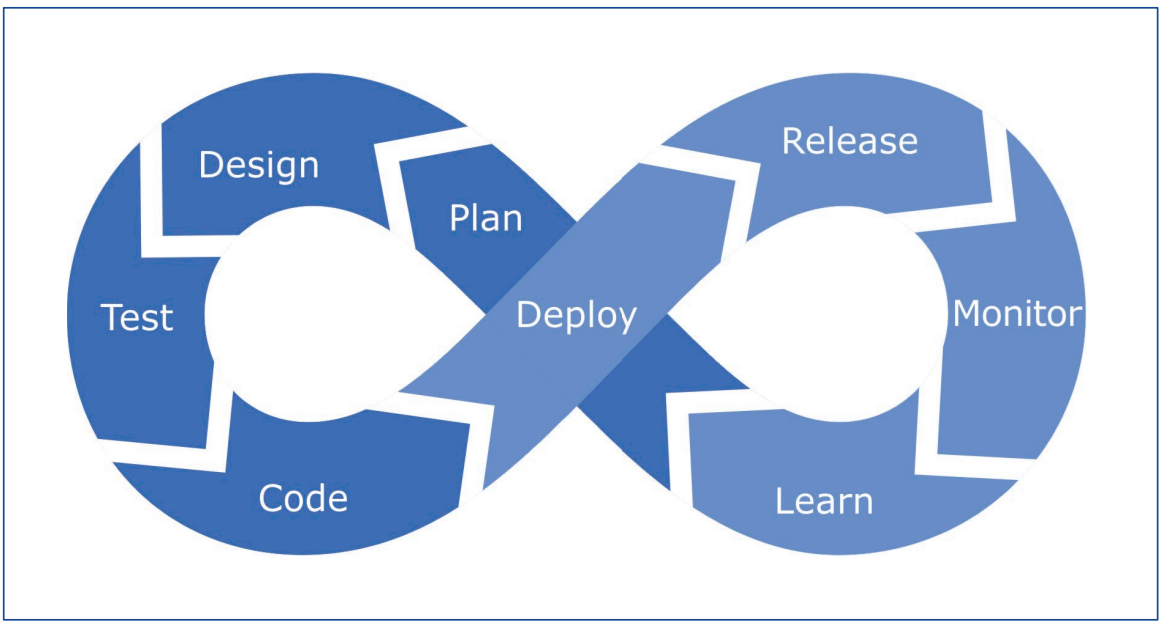


Figure 1-1, DevOps Lemniscate.

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 security. Continuous security is not one isolated aspect area of the Continuous Everything (CE) concept but affects 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)	6	Continuous Deployment (Release)
2	Continuous Design (Design)	7	Continuous Monitoring (Monitor)
3	Continuous Testing (Test)	8	Continuous Learning (Learn)

Development		Operations	
4	Continuous Integration (Code)	9	Continuous Security (-)
5	Continuous Deployment (Deploy)	10	Continuous Assessment (-)

Table 1-1, Continuous everything aspects.

Continuous auditing (9) and continuous assessment (10) are not explicitly reflected in the DevOps Lemniscate, as are other continuous aspect areas such as continuous documentation and continuous robotics for the sake of simplicity of the DevOps Lemniscate.

The word "continuous" refers to the incremental and iterative development of software, creating a "stream" of code that is continuously transitioned to production through the CI/CD secure pipeline. This 'stream' is nothing but a value stream that needs to be optimised.

Continuous security is an Agile way of realising the required controls during the design of an information system by defining the behavior, functionality, and quality of the information system at the right time that mitigate or eliminate the identified information security risks. All aspects of the DevOps Lemniscate have a direct or indirect relationship with continuous security because it is designed holistically. This means that continuous security relates to both the information system (Technology) and the production process (Process) as well as knowledge and skills (People). With this, continuous security provides a design at PPT level.

An important basis for securing the controls of an information system is its design. In recent years, however, many organisations have questioned the existence of a design for an information system. The classic justification of bundling information about an information system and involving all stakeholders is seen as outdated by the Agile way of working and the idea of the three-amigo development strategy. This strategy means that from three disciplines: business, development and testing, an increment that needs to be built is examined in advance. In this way, the 'how' and the 'what' questions are better worked out and consensus can be reached about the Definition of Done (DoD) of the increment. However, this ignores the other classic justification of a design, which is that a design is also intended for the control function that prevents information security risks from materialising due to a lack of countermeasures. These risks concern various aspects, including non-compliance with legal and regulatory obligations. Another important aspect of controls are those of information security such as security, integrity, and confidentiality of information.

From the point of view of continuous security, the developments taking place within the DevOps world are therefore very important to follow. While on the one hand there are currently still organisations that work with waterfall projects that require a large design effort, there are also organisations that experience that working with user stories alone is not the ultimate solution and that some form of design is indeed necessary. is. And so, the world of system development comes into balance again and continuous security is given a basis.

The question is, of course, whether the same structure of work should apply to all types of information systems. With the arrival of Gartner's BI model, it has become clear that a distinction must be made between the System of Records (SoR) and the System of Engagement (SoE) information systems. In addition to the SoE, people nowadays also talk about the System of Intelligence (SoI). Figure 1-2 provides an overview of the relationship between the three types of information systems (SoR, SoE and SoI).

System of Records

The SoR are information systems of the back office that fulfil the finance, logistics, inventory, and Human Resource Management (HRM) tasks. These systems must meet the requirements of information security in terms of Confidentiality, Integrity, and Availability (CIA). This means, among other things, that designs are required that indicate how the financial data is generated and what the interfaces are of the different involved information systems. They are generally information systems that are part of a chain of information systems. These systems require a well-considered approach and therefore a design in which information security plays an important role.

System of Engagement

The SoE information systems are aimed at sales channels to consumers, especially web shops and apps for smartphones.

These applications are easy to provide with a new release, version, and patch. These information systems are usually not an integral part of a chain but rather the end points of a chain. These are often also the examples from the publications about Agile and Development & Operations (DevOps). For these information systems it is clear that a design that has been thought through in advance (upfront design) is less necessary and can often suffice with a growing design (emerging design). However, it is important to consider these endpoints as possible leaks in information security. That is why it is useful for these SoE information systems to have more than just a collection of user stories. Information security must also form an integral part of these information systems. The separate user stories also do not form an accessible description of the information security of an information system. There is therefore still a need for a design that provides an overview and insight into the functionality, quality, and operation of the information system as well as information security. In particular, the user interface and interfaces with data sources are important aspects of security and therefore continuous security.

System of Intelligence

In addition, there are Business Intelligence (BI) solutions. These are the reports, data analysis tools and the like. The same applies to these applications as the SoE information systems. They are the representation of information from the SoR and are easier to modify. However, data leakage and intrusion are also risks. Depending on the value of the information that is made accessible, the risk is higher or lower. Information must therefore also be known about the risks and what countermeasures must be taken.

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

Need for continuous security

For all three types (SoR, SoE and SoI) information systems there is therefore a need for a certain degree of control and therefore for continuous security.

More than just a set of user stories is needed to gain and maintain an overview and insight into the control requirements imposed on the information system. Otherwise, the risks and impact of adapting and expanding the information system cannot be recognised in a timely and correct manner. However, it must be prevented that the implementation of countermeasures (controls) of the risks does destroy the Agility of the production process. This means that controls must not only be designed and monitored incrementally and iteratively (continuous security), but that the controls must also be selected on the basis of the weighted risk, i.e. not too much, but certainly not too little. The extent to which the design of the control must be defined is shifting from a lot with SoR to less with SoE and almost nothing with SoI. But also with a SoR, the design of controls can be divided into layers that are defined more upfront (in advance) and emerging (in sprints).

1.4 Structure

This book discusses how to shape continuous security using the Information Security Value System (ISVS) model. Before discussing this model, the definitions, anchoring and architecture of continuous security are first given substance. This is followed by a discussion of this model.

1.4.1 *Chapter 2: Basic concepts and basic terms*

This chapter discusses the basic concepts and the basic terms.

1.4.2 *Chapter 3: Continuous security definition*

It is important to have a common definition of continuous security. Therefore, this chapter defines this concept and discusses the problems and possible causes of improper design and management of an information system.

1.4.3 *Chapter 4: Continuous security anchorage*

This chapter discusses how continuous security can be anchored through the change paradigm. The following questions are answered.

- What is the vision on continuous security (Vision)?
- Where do the responsibilities and authorities lie (Power)?
- How can continuous security be applied (Design)?
- Which profiles of people and which resources are needed (Resources)?

1.4.4 *Chapter 5: Continuous security architecture*

This chapter describes the architecture principles and models for continuous security. The architectural models concern the following models:

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

1.4.5 *Chapter 6: Continuous security design*

The ISVS consists of an information security value chain that gives substance to the information security value streams. The value streams are described based on use cases. The relationships between the use cases are shown in a use case diagram.

1.4.6 *Chapter 7: Continuous security best practices*

This chapter describes how to construct the ISVS and discusses a number of continuous security best practices.

1.4.7 *Chapter 8: Governance security practices*

This chapter discusses governance security practices. These are the best practices that govern the ISVS value chain. It concerns the practices for obtaining top management commitment, determining interested parties, determining the scope, determining goals, and determining the information security policy. For each security practice, the use case is discussed, as well as the definition, the objective, and an example of which application.

1.4.8 *Chapter 9: Risk security practices*

The risk security practices consist of the operational best practices of the ISVS. This concerns the practices for determining the issues, risk criteria and information assets, identifying risks, performing a risk assessment and risk treatment as well as realising controls. For each security practice, the use case is discussed, as well as the definition, the objective, and an example of which application.

1.4.9 *Chapter 10: Quality security practices*

The quality security practices consist of the internal audit and the continuous improvement of best practices of the ISVS. For each security practice, the use case is discussed, as well as the definition, the objective, and an example if applicable.

1.4.10 *Chapter 11: Continuous security versus Agile Scrum*

Approaching continuous security in isolation is pointless. There are too many interfaces with system development (Agile Scrum) so that the integration of the disciplines is necessary. This chapter describes the essence of this link.

1.4.11 *Chapter 12: Continuous security versus DevOps*
The link between information security and Agile Scrum is a very important improvement. However, a link with operations is also required to operationalise continuous security. This chapter discusses how to integrate continuous security with DevOps using the concept of continuous everything.

1.4.12 *Chapter 13: Continuous security assessment*
The maturity of continuous security has been made measurable in this chapter based on a continuous security assessment.

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 spelled out.
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 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.
Best 2022a	B. de Best, "Continuous Development", English language, Leonon Media, 2022, ISBN13: 978 94 92618 764.

References	Publications
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 Organizations, 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.
scrum	Ken Schwaber and Jeff Sutherland, "The Scrum Guide™", 2017, www.scrumguides.org.

References	Publications
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 (躾): Hold or Systematise [Wiki]
A/B testing	A/B testing means that two versions of an application or webpage are taken into production to see which performs better. Canary releasing can be used, but there are also other ways to perform A/B testing.
Acceptance test	For DevOps engineers the acceptance testcases gives the answer "How do I know when I am done?". For the users the acceptance testcases gives the answer "Did I get what I wanted?". Examples of acceptance testcases are Functional Acceptance Testcases (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 Given – When – Then format.
Binary	A compiler is used to transform source code to object code. The object code is also known as a binary. The source code is readable for human being, the object code however is only readable for computers since they have been written in hexadecimals.
Blameless post-mortem	Blameless post-mortem is a term coined by John Allspaw. It helps to examine "mistakes in a way that focuses on the situational aspects of a failure's mechanism and the decision-making process of individuals proximate to the failure." [Kim 2016].
Blamelessness	This approach is about learning rather than punishing. Within DevOps this is one of the basic ideas of learning from mistakes. The energy of the DevOps team is spending on learning from the mistake, rather than on finding the one to blame.
Blue-Green deployment pattern	Blue and green refer to two identical production systems. One is used for the final acceptance of a new release. If this acceptance is successful, then this environment becomes the new production environment. In case of a failure of the production system, the other system can be used instead. This mitigates the risk of downtime since the switchover is likely to be less than a second.
Broken build	A build that fails due to an error in the application source code.
Brown field	There are two scenarios' for applying DevOps best practices: green field and brown field. In case of a green field scenario the whole DevOps organisation has to be established from scratch. The opposite scenario is where there is already a DevOps organisation, but improvements are needed. The colour green refers to the situation that a factory is built on a clean grass field.

Term	Meaning
	The colour brown refers to the situation that a factory is to be built on a place where there has already been a factory that poisoned the ground. In order to build on a brown field, the poison needs to be removed.
Business value	Applying DevOps best practices results in increasing the business value. Research of Puppet Labs (State Of DevOps Report) proofs that high-performing organisations using DevOps practices are outperforming their non–high performing peers in many following areas [Kim 2016].
Canary releasing pattern	Normally a release is offered to every user at once. Canary releasing is the approach in which a small set of users is receiving the new release. If this small scope release works fine than the release can be deployed to all users. The term canary refers to the old habit to have a canary in the coal mines to detect toxic gas.
Change categories	Changes can be categorised into standard changes, normal changes and urgent changes.
Change schedules	Changes can be scheduled in order to defined in which order they have to be applied.
Cloud configuration files	Cloud configuration files are used to initiate a cloud service before using it. In this way cloud service providers enable customers to configure the cloud environment for their needs.
Cluster immune system release pattern	The cluster immune system expands upon the canary release pattern 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 branching.
Code review methods	Code review can be performed in several ways like "over the shoulder", pair programming, email pass-around and tool-assisted code review.
Codified NFR	A list of Non-Functional Requirements (NFR) that are categorised in categories like availability, capacity, security, continuity et cetera.
Collaboration	One of the four pillars of DevOps is collaboration. Collaboration refers to the way the individuals of a DevOps team works together to achieve the common goal. There are many forms in which this collaboration comes to expression like: • peer to peer programming; • demonstrating weekly progress; • documentation; et cetera.
Commit code	Committing code is the action in which the DevOps engineer adds the changed source code to the repository, making these changes part of the head revision of the repository [Wiki].
Commit stage	This is the phase in the CI/CD secure pipeline where the source code is compiled to the object code. This includes the performance of the unit testcases.
Compliance checking	The manual action of a security officer to make sure that the system is built in accordance with the agreed standards.

Term	Meaning
	This is the opposite of security engineering where the DevOps teams works together with the security officer in order to embed the agreed standards in the deliverables and enable continuous monitoring of the standard in the whole lifecycle of the product.
Compliancy officer	The compliancy officer is a DevOps role. The compliancy officer is responsible for ensuring compliance with agreed standards throughout the whole life cycle of a product.
Configuration management	Configuration Management refers to the process by which all artefacts, and the relationships between them, are stored, retrieved, uniquely identified and modified.
Containers	A container is an isolated structure that is used by DevOps engineers to build their application independently from the underlying operating system or hardware. This is accomplished by interfaces in the container that are used by DevOps engineers. Instead of installing the application in an environment, the complete container is deployed. This saves a lot of dependencies and prevents configuration errors to occur.
Conway's law	The following statement of Melvin Conway is called the Conway's law: "organisations which design systems ... are constrained to produce designs which are copies of the communication structures of these organisations." [Wiki].
Cultural debt	There are three forms of debt. Cultural debt, technical debt and information debt. This form of debt refers to the decision to keep flaws in the organisation structure, hiring strategy, values et cetera. This debt costs interest and will result in less maturity growth of the DevOps teams. Cultural debt can be recognised by the exitance of extensive silos, workflow constraints, miscommunications, waste et cetera.
Culture, Automation Measurement, Sharing (CAMS)	CAMS is the abbreviation for Culture, Automation, Measurement and Sharing. • Culture: Culture 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 programming paradigm that expresses the logic of a computation without describing its control flow. An example are the database query languages for example TSQL and PSQL.

Term	Meaning
Defect tracking	Defect tracking is the process of tracking the logged defects in a product from beginning to closure and making new versions of the product that fix the defects [Wiki].
Development	Development is an activity that is performed by the DevOps role 'DevOps engineer'. A DevOps engineer is responsible for the complete lifecycle of a configuration item. Within DevOps there is no difference anymore between designer, builder or tester.
Development rituals	The Agile Scrum rituals of development are the sprint planning, daily stand-up, sprint execution, review and the retrospective.
Downward spiral	Gene Kim explains in his book [Kim 2016] that the downward spiral in Information Technology (IT) has three acts. • The first act begins in IT Operations where technical debt results in jeopardising our most important organisational promises. • The second act starts with compensating the latest broken promise by promising a bigger, bolder feature or an even larger revenue target. As a result, Development is tasked with another urgent project which results in even more technical debt. • The third stage is where the deployments are getting slower and slower, and outages are increasing. The business value continuously decreases.
E-mail pass-around	E-mail pass-around is a review technique where the source code management system emails code to reviewers automatically after the code is checked in [Kim 2016].
Error path	See happy path.
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 programming paradigm 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 declarative programming, which focuses on what the program should accomplish without specifying how the program should achieve the result.
Independent, Negotiable, Valuable, Estimable, Small, and Testable (INVEST)	Independent, Negotiable, Valuable, Estimable, Small, and Testable. • Independent: The product backlog item should be self-contained, in a way that there is no inherent dependency on another product backlog item. • Negotiable: Product backlog items, up until they are part of an iteration, can always be changed, rewritten or even discarded. • Valuable: Product backlog item must deliver value to the stakeholders.

Term	Meaning
	• Estimable: The size of a product backlog item must always estimable. • Small: Product backlog items should not be so big as to become impossible to plan / task / prioritise with a certain level of certainty. • Testable: The product backlog item or its related description must provide the necessary information to make test development possible.
Information radiators	An Information Radiator is a visual display that a team places in a highly visible location so that all team members can see the latest information at a glance.
Infosec	A team that is responsible for securing systems and data.
Infrastructure as Code (IaC)	Normally infrastructure components have to be configured in order to perform the requested functionality and quality for example a rule set for a firewall or the allowed IP addresses for a network. These configurations normally are stored in configuration files which enable the operators to manage the functionality and the quality of the infrastructure components. Infrastructure as code (IaC) makes it possible to programme these infrastructure component settings and deploy these settings through the CI/CD secure pipeline by the use of machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.
Infrastructure as Code (IaC)	Infrastructure as code (IaC) is a software-based approach to the ICT infrastructure, whereby the systems can be rolled out and adapted in a consistent manner through templates. If a change has to be made, it is implemented in the template which is then rolled out again.
Infrastructure management	Infrastructure management consists of the lifecycle management of all infrastructure products and services in order to support the correct working of the applications that run on top of the infrastructure.
Ji-Kotei-Kanketsu (JKK)	JKK which means 100% completion of an item. This quality way of working means: • clear understanding of the goals; • understanding the right way to work; • ensure high quality of work; • getting the work right for 100% completion, never pass defects to the next process; • Definition of Done (DoD) is vital; and then maintaining the required quality without inspections.
Just In Time (JIT)	JIT means building up a stream-lined supply chain with one-piece flow.
Kaizen	Kaizen is Japanese for “improvement”. Kaizen is used to improve production systems. The goals of kaizen are: • elimination of waste (muda's); • JIT; • standardisation of production; • cycle of continuous improvements. Continuous improvement means circulate the Plan-Do-Check-Act (PDCA) cycle daily, weekly. This can be accomplished by finding the root cause of a failure by asking "Why" 5 times. The following steps can be followed: • defining problems with supporting data; • making sure everybody recognises the problems clearly;

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

Term	Meaning
	• 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	A learning culture is a collection of organisational conventions, values, practices and processes. These conventions encourage employees and organisations to develop knowledge and competence. An organisation with a learning culture encourages continuous learning and believes that systems influence each other. Since constant learning elevates an individual as a worker and as a person, it opens opportunities for the establishment to transform continuously for the better.
Light weight ITSM	This variant of Information Technology (IT) Service Management (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.
Logging levels	Within monitoring systems there are several levels of logging recognised: • Debug level: Information at this level is about anything that happens in the program, most often used during debugging. • Info level: Information at this level consists of actions that are user-driven or system specific. • Warn level: Information at this level tells us of conditions that could potentially become an error. • Error level: Information at this level focuses on error conditions Fatal level: Information at this level tells us when we must terminate.
Loosely coupled architecture	Loosely coupled architectures enables that changes can be made safely and with more autonomy, increasing developer productivity.
Micro service	Microservices are a variant of the service-oriented architecture (SOA) architectural style that structures an application as a collection of loosely coupled services. In a microservices architecture, services should be fine-grained, and the protocols should be lightweight [Wiki].
Micro service architecture	This architecture consists of a collection of services where each service provides a small amount of functionality, and the total functionality of the system is derived from composing multiple versions of a service in production simultaneously and to roll back to a prior version relatively easily.
Mini pipeline	In rare cases more than one deployment pipeline is required in order to produce the entire application. This can be accomplished by the use of a pipeline per application component. All these components are then assembled in a central pipeline which puts the entire application through acceptance tests, non-functional tests, and then deploys the entire application to testing, staging, and production environments.

Term	Meaning
Monitoring Framework	A framework of components that together form a monitor facility that is capable to monitor business logic, applications, and operating systems. Events, logs and measures are routed by the event router to destinations [Kim 2016].
Monolithic	A monolithic architecture is the traditional programming model, which means that elements of a software program are interwoven and interdependent. That model contrasts with more recent modular approaches such as a micro service architecture (MSA).
MTTR	Mean Time To Repair (MTTR) is a basic measure of the maintainability of repairable items. It represents the average time required to repair a failed component or device.
Muda	This is a Japanese word for waste. It is used in relationship to production systems.
Non-Functional Requirement (NFR)	NFR are requirements that define the quality of a product like maintainability, manageability, scalability, reliability, testability, deploy ability and security. NFR are also referred to as operational requirements.
Non-Functional Requirement (NFR) testing	NFR testing is the testing aspect that focusses on the quality of the product.
Obeya	Obeya is a war room which serves two purposes: • information management; • and on-the-spot decision making.
One piece flow	The Lean approach means that the DevOps team only works at one item at a time as a team with a fast pace and smooth flow. This is also used in the first way of the three ways of Gene Kim.
Operations	Operations is the team often responsible for maintaining the production environment and helping to ensure that required service levels are met [Kim 2016].
Operations stories	The work that has to be done by Ops can be written in stories. In that way that can be prioritised and managed.
OPS liaison	An OPS liaison is an operation employee who is assigned to a development team in order to facilitate the development team for their infrastructural demands.
Organisation archetypes	There are three organisation archetypes: functional, matrix, and market. They are defined by Dr. Roberto Fernandez as follows: • Functional: Functional-oriented organisations optimise for expertise, division of labour, or reducing cost. • Matrix: Matrix-oriented organisations attempt to combine functional and market orientation. • Market: Market-oriented organisations optimise for responding quickly to customer needs.
Organisational typology model	This a model of Dr. Ron Westrum in which he defined three types of culture: 'pathological', 'bureaucratic', 'generative'. These organisation types can be recognised by the following characteristics: • Pathological organisations are characterised by large amounts of fear and threat. • Bureaucratic organisations are characterised by rules and processes. • Generative organisations are characterised by actively seeking and sharing information to better enable the organisation to achieve its mission.

Term	Meaning
	Dr. Westrum observed that in healthcare organisations, the presence of "generative" cultures was one of the top predictors of patient safety.
Over-the-shoulder	This is a review technique where the author walks through his code while another developer gives feedback.
Packages	A set of individual files or resources which are packed together as a software collection that provides certain functionality as part of a larger system.
Pair-programming	This is review technique where two developers work together using one computer. While one developer writes the code the other reviews it. After one hour they exchange their role.
Peer review	This is a review technique where developers review each other's code.
Post-mortems	After a major incident a post-mortem meeting can be organised in order to find out what the root-cause is of the incident and how to prevent it in the future.
Product owner	The Product Owner is a DevOps role. The Product Owner is the internal voice of the business. The Product Owner is the owner of the product backlog and determines the priority of the product backlog items in order to define the next set of functionalities in the service.
Programming paradigm	A style of building the structure and elements of computer programs.
Pull request process	This is a form of peer review that span Dev and Ops. It is the mechanism that lets engineers tell others about changes they have pushed to a repository.
Quality Assurance (QA)	Quality Assurance (QA) is the team responsible for ensuring that feedback loops exist to ensure the service functions as desired [Kim 2016].
Reduce batch size	The size of a batch has an influence on the flow. Small batch sizes results in a smooth and fast flow. Large batch sizes results in high Work In Progress (WIP) and increases the level of variability in flow.
Reduce number of handoffs	In terms of a software process a handoff means that the work that is performed in order to produce software is stopped and handed over to another team. Each time the work passes from one team to another team, this requires all sorts of communication using different tools and filling up queues of work. To less handoffs the better.
Release managers	This a DevOps role. The release manager is responsible for managing and coordinating the production deployment and release processes.
Release patterns	There are two patterns of releases to be recognised [Kim 2016]: • Environment-based release patterns: In this pattern there are two or more environments that receive deployments, but only one environment is receiving live customer traffic. • Application-based release patterns: In this pattern the application is modified in order to make selectively releases possible and to expose specific application functionality by small configuration changes.
Sad path	A specific type of a 'bad path' 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 HRR of an LRR.

Term	Meaning
SBAR	This technique offers guidelines for making sure concerns or critiques are expressed in a productive manner. In this situation the people who concerns it have to follow the following steps: • situational information to describe what is happening; • background information or context; • an assessment of what they believe the problem is; • recommendations for how to proceed.
Security testing	Security testing is one of many types of tests. Within DevOps security testing is integrated in the deployment pipeline by using automated tests as early as possible in the flow.
Self service capability	One way of integrating Ops in Dev is the usage of infrastructure self-services.
Shared goals	Delivering value to the customer requires that Dev and Ops are working together in value streams and have shared goals and practices.
Shared Operations Team (SOT)	A SOT is a team that is responsible for managing all the DTAP environments performing daily deployments into those development and test environments, as well as doing periodically production deployments. The reason to use a SOT is to have a team that focusses only on deployments. This results in automation of repeatable work and learning how to fix occurring problems very fast.
Shared version control repository	In order to be able to use trunk-based development DevOps engineers need to share their source code. The source code must be committed into a single repository that also supports version control. Such a repository is called a shared version control repository.
Simian army	Simian Army consists of services (Monkeys) for generating various kinds of failures, detecting abnormal conditions, and testing the ability to survive them. The goal is to keep the cloud service safe, secure, and highly available. Currently there are 3 Monkeys in the Simian Army: • Janitor Monkey (unused resources); • Chaos Monkey (try to shut down a service); • Conformity Monkey (non-conformance to rules).
Single repository	A single repository is used to facilitate trunk-based development.
Smoke testing	Smoke testing is one of the test types that is used to determine whether or not the basics of a new or adjusted service works. Only a few testcases are needed to indicate whether or not at least the most important functions are working properly. This test type origins from the hardware manufacturers where engineers tested circuits by powering on the system and checking for smoke which was an alarm of malfunctioning hardware.
Standard deviation	In statistics, the standard deviation (SD, also represented by the Greek letter sigma σ or the Latin letter s) is a measure that is used to quantify the amount of variation or dispersion of a set of data values. A low standard deviation indicates that the data points tend to be close to the mean (also called the expected value) of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values [Wiki].
Standard operations	The standard operations is the situation in which the system performs as designed. Deviations of the standard operations need to be detected as early as possible.

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

Term	Meaning
	The charter and the principles formed an elaboration of ideas that had arisen in the mid-nineties, in response to methods traditionally classed as waterfall development models. Those models were experienced as bureaucratic, slow, and narrow-minded and would hinder the creativity and effectiveness of DevOps engineers. The seventeen people who have drawn up the Agile Manifesto together represented the various Agile movements. After the publication of the charter, several signatories set up the "Agile Alliance" to further convert the principles into methods [Wiki].
The ideal testing automation pyramid	The ideal testing automation pyramid is a way of testing that can be characterised as follows: • Most of the errors are found using unit tests as early as possible. • Run faster-running automated tests (e.g., unit tests) before slower-running automated tests (e.g., acceptance and integration tests), which are both run before any manual testing. • Any errors should be found with the fastest possible category of testing.
The Lean movement	An operating philosophy that stresses listening to the customer, tight collaboration between management and production staff, eliminating waste and boosting production flow. Lean is often heralded as manufacturers' best hope for cutting costs and regaining their innovative edge.
The non-ideal testing automation inverted pyramid	The non-ideal testing automation pyramid is a way of testing that can be characterised as follows: • Most of the investment is in manual and integration testing. • Errors are found later in the testing. • Slower running automated tests are performed first.
The Simian Army	The Simian Army is a collection of open-source cloud testing tools created by the online video streaming company, Netflix. The tools allow engineers to test the reliability, security, resiliency and recoverability of the cloud services that Netflix runs on Amazon Web Services (AWS) infrastructure [Whatis]. Within this Simian Army the following monkeys are recognised: Chaos Gorilla, Chaos Kong, Conformity Monkey, Doctor Monkey, Janitor Monkey, Latency Monkey and Security Monkey.
The three ways	The three ways are introduced in 'The Phoenix Project: A Novel About IT, DevOps, And Helping Your Business Win' by Gene Kim, Kevin Behr and George Spafford. The Three Ways are an effective way to frame the processes, procedures and practices of DevOps, as well as the prescriptive steps. • The first way – flow understand and increase the flow of work (left to right); • The second way – feedback create short feedback loops that enable continuous improvement (right to left); • The third way – Continuous Experimentation and Learning (continuous learning).
Theory of constraints	This is a methodology for identifying the most important limiting factor that stands in the way of achieving a goal and then systematically improving that constraint until it is no longer the limiting factor.
Tool-assisted code review	This is a review technique where authors and reviewers use specialised tools designed for peer code review or facilities provided by the source code repositories [Kim 2016].

Term	Meaning
Toyota Kata	Toyota Kata is a management book by Mike Rother. The book explains the Improvement Kata and Coaching Kata, which are a means for making the Continual improvement process as observed at the Toyota Production System teachable [Wiki].
Transformation team	Introducing DevOps requires a defined transformation strategy. Based on their research, Dr. Govindarajan and Dr. Trimble assert that organisations need to create a dedicated transformation team that is able to operate outside of the rest of the organisation that is responsible for daily operations (which they call respectively the "dedicated team" and "performance engine"). The lessons learned from this transformation team can be used to apply in the rest of the organisation.
Value stream	The process required to convert a business hypothesis into a technology-enabled service that delivers value to the customer [Kim 2016].
Value Stream Mapping (VSM)	Value stream mapping is a Lean tool that depicts the flow of information, materials, and work across functional silos with an emphasis on quantifying waste, including time and quality.
Vertical splitting of features	A feature can be splitted into stories. Vertical splitting refers to the result of a feature splitting in which more DevOps teams can work independently on their own stories. Together they realise the feature. See also Horizontal splitting of features.
Virtualised environment	An environment that is based on virtualisation of hardware platforms, storage devices and network resources. In order to create a virtualised environment usually VMware is used.
Visualisation	In computing, virtualisation refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources. Virtualisation began in the 1960s, as a method of logically dividing the system resources provided by mainframe computers between different applications. Since then, the meaning of the term has broadened [Wiki].
Walking skeleton	Walking skeleton means doing the smallest possible amount of work to get all the key elements in place.
Waste	Waste comprises the activities that are performed in the manufacturing process that are not adding value to the customer. Examples in the context of DevOps are: • Unnecessary software features. • Communication delays. • Slow application response times. • Overbearing bureaucratic processes.
Waste reduction	Minimisation of waste at its source is to minimise the quantity required to be treated and disposed of, achieved usually through better product design and/or process management. Also called waste minimisation [Businessdictionary].
WIP limit	This is a Key Performance Indicator (KPI) that is used in the Kanban process to maximise the number of items that has been started but that is not completed. Limiting the amount of WIP is an excellent way to increase throughput in your software development pipeline.
Work In Progress (WIP)	Material that has entered the production process but is not yet a finished product.

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
AWS	Amazon Web Services
BDD	Behavior Driven Development
BI	Business Intelligence
BOK	Body of Knowledge
BSC	Balanced Score Card
BVS	Business Value System
CA	Competitive Advantage
CA	Continuous Auditing
CAB	Change Advisory Board
CAMS	Culture, Automation, Measurement and Sharing
CD	Continuous Deployment
CE	Continuous Everything
CEM	Central Event Monitor
CEMLI	Configuration, Extension, Modification, Localisation, Integration
CEO	Chief Executive Officer
CFO	Chief Finance Officer
CI	Configuration Item
CI	Continuous Integration
CIA	Confidentiality, Integrity & 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
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
CY	Continuous securitY
DevOps	Development & Operations
DML	Definitive Media Library

Abbreviation	Meaning
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
JIT	Just In Time
JKK	Ji-Kotei-Kanketsu
JVM	Java Virtual Machine
KPI	Key Performance Indicator
LAN	Local Area Network
LCM	LifeCycle Management
LDAP	Lightweight Directory Access Protocol

Abbreviation	Meaning
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
RUM	Real User Monitoring
S-CI	Software Configuration Item
SA	Strategic IS Architecture
SAFe	Scaled Agile Framework
SAT	Security AcceptatieTest
SBAR	Situation, Background, Assessment, Recommendation
SBB	System Building Block
SBB-A	System Building Block Application
SBB-I	System Building Block Information
SBB-T	System Building Block Technology

Abbreviation	Meaning
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
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-publications_2021-07-22_900.pdf
De Caluwé	[De Caluwé]	https://www.agile4all.nl/het-kleurenmodel-van-de-caluwe-en-vermaak/
DevOps	[DevOps]	http://DevOps.com
DDD	[DDD]	https://www.slideshare.net/skillsmatter/ddd-in-agile
doxygen	[doxygen]	http://www.doxygen.nl/manual/docblocks.html
doxygen example	[doxygen example]	http://www.doxygen.nl/manual/examples/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 · 10, 56, 58, 61, 63, 65, 70, 74, 75, 77, 78, 83, 91, 94, 97, 99, 102, 104, 106, 110, 113, 115, 117, 118, 119, 121, 122, 175

A

A/B testing · 159
abuse of audit tools · 79
abuse of information system · 79
acceptance criteria · 45, 51, 82, 83, 84, 86, 93, 116, 130, 139
acceptatiecriterium · 160, 164
acceptatietest · 159
account · 81, 88
actor · 48, 50, 56, 59, 61, 63, 64, 65, 66, 70, 75, 78, 79, 83, 91, 92, 94, 97, 99, 102, 104, 106, 110, 113, 115, 116, 117, 118, 119, 122, 123
adaptive software development · 128
added value · 32, 53, 63, 126, 127, 128, 130, 131
affinity · 159
Agile · 159, 171, 172
Agile infrastructure · 159
Agile modeling · 128
Agile Scrum · 4, 5, 12, 15, 24, 25, 27, 43, 125, 128, 129, 130, 132, 133, 134, 135, 136, 137, 163
Agile Scrum process · 24, 125
Agile unified process · 128
alternate path · 159
Amazon Web Services · See AWS
Andon cord · 159
anomaly detection technique · 159
antifragility · 160
anti-pattern · 22, 23, 26, 27, 29, 159
applicatiebeheer · 166
applicatiecomponent · 167
artefact · 160, 162
artefact repository · 160
assessment · 170
asset · 45, 51, 72, 91, 92, 148, 149
- category · 45
- group · 45
- inventory · 85
- register · 45, 63, 93
auditor · 1
automated test · 160
availability · 65, 87, 88, 89, 90, 161
awareness training · 51, 58, 65, 66
AWS · 175

B

backlog item · 169
bad apple theory · 160
bad path · 160
Balanced Score Card · See BSC
BDD · 160, 175
Behavior Driven Development · See BDD
best practice · 161
BI · 2, 3, 175
binary · 160
blameless post mortem · 160
blamelessness · 160
blue/green deployment · 160
Body of Knowledge · See BOK
BOK · 175
branching · 161
breach · 80
broken build · 160
brown field · 160
BSC · 8, 175
build · 160, 161, 162, 171
business
- case · 17, 21, 26, 66, 139, 140
- impact · 73, 74, 77
- process · 10, 127
- value chain · 12, 15
Business Intelligence · See BI
business value · 161, 163
Business Value System · See BVS
BVS · 12, 13, 33, 38, 175

C

CA · 175
CAB · 175
CAMS · 162, 175
canary releasing · 161
capability · 162
Capability Maturity Model Integration · See CMMI
capaciteit · 161
CCTA Risk Assessment Method Methodology · See CRAMM
CD · 146, 161, 165, 175
CE · 175
CEM · 175
CEMLI · 175
CE-model · 145
Central Event Monitor · See CEM
Central Processing Unit · See CPU
CEO · 56, 57, 58, 175
CFO · 56, 57, 58, 175
Change Advisory Board · See CAB
change category · 161
change paradigm · 4, 21, 22, 23, 24, 27, 28, 31, 139, 201

change schedule · 161
Chief Executive Officer · See CEO
Chief Finance Officer · See CFO
Chief Information Officer · See CIO
Chief Technology Officer · See CTO
CI · 145, 146, 161, 165, 175
CI/CD secure pipeline · 2, 12, 24, 64, 67, 111, 112, 139, 140, 142, 149
CIA · 2, 14, 45, 46, 50, 63, 64, 65, 67, 69, 70, 71, 72, 74, 78, 79, 81, 82, 84, 88, 89, 90, 93, 94, 95, 96, 97, 98, 101, 103, 104, 105, 106, 108, 111, 112, 116, 120, 125, 134, 135, 137, 175
CIA-matrix · 14, 46, 63, 65, 67, 69, 78, 82, 93, 94, 96, 101, 104, 105, 106, 108, 111, 112, 116, 125, 137
CIO · 175
CISO · 56, 57
CL · 146, 175
cloud · 161
cloud configuration file · 161
cloud provider · 59
cloud service · 17, 22, 161
cluster immune system release pattern · 161
CM · 146, 150, 175
CMDB · 175
CMMI · 147, 175
CMS · 175
CN · 175
CO · 146, 175
CoC · 45, 66, 67, 175
code branch · 161
Code of Conduct · 51, 65, 66, 67, 68, 72, 91, See CoC
code review form · 161
codified NFR · 161
collaboration · 161
commit code · 161
commit stage · 161
commitment statement · 45, 56, 57
Communities of Practice · See CoP
competence · 159, 164
Competitive Advantage · See CA
Competitive Response · See CR
Completeness / Accurateness · See %C/A
compliance · 1, 17, 18, 27, 32, 33, 38, 57, 66, 68, 105, 114, 139, 146, 162
compliance checking · 162
compliancy · 162
compliancy officer · 162
component · 165, 168, 171
confidentiality · 2, 14, 23, 64, 88, 89, 90
Confidentiality, Integrity & Accessibility · See CIA
Configuration Item · See CI
configuration management · 162
Configuration Management DataBase · See CMDB
Configuration Management System · See CMS
Configuration, Extention, Modification, Localisation, Integration · See CEMLI
container · 162
continuity · 161, 167
continuous
- assessment · 2, 138, 142
- auditing · 142
- deployment · 1, 2, 138, 142
- design · 1, 34, 138, 139, 140, 141
- everything · 146, 147
- improvement · 172
- integration · 2, 128, 138, 142
- learning · 1, 138, 142, 172
- monitoring · 1, 138, 142
- planning · 1, 138, 141
- security · 1, 2, 3, 17, 46, 47, 48
- security assessment · 5
- security pyramid · 7, 12, 21, 22, 24, 25, 31, 34, 35, 36, 41, 42, 43, 138
- testing · 1, 138, 142

Continuous aSsessment · See CS
Continuous Auditing · See CA
continuous control model · 37
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 · See CP
Continuous securitY · See CY
Continuous Testing · See CT
contract · 126
control · 2, 3, 4, 7, 8, 9, 12, 13, 15, 17, 18, 19, 22, 23, 24, 25, 26, 27, 28, 32, 33, 34, 36, 37, 38, 40, 42, 43, 49, 52, 53, 57, 62, 63, 69, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 117, 121, 125, 133, 134, 135, 136, 137, 141, 142, 143,148, 149, 150, 162, 170
- backlog · 15
- evidence database · 46
- lifecycle management · 13
- requirements · 3

Conway's law · 162
cookbook · 73, 77, 82, 96, 98, 101, 103, 107, 112, 115
CoP · 25, 175
corporate espionage · 79, 81
counter measure · 166
countermeasure · 2, 3, 8, 13, 14, 17, 40, 42, 49, 55, 63, 64, 65, 67, 99, 101, 105, 106, 107
CP · 175
CPU · 175
CR · 175
CRAMM · 175
CRAMM issue · 14, 79
CRAMM issue register · 45
CRAMM threat · 80
CRC · 175
CRC code · 64
Critical Success Factor · See CSF

CS · 175
CSF · 45, 55, 63, 64, 65, 67, 70, 175
CSI register · 46, 124
CT · 145, 175
CTO · 175
cultural debt · 162
Culture, Automation, Measurement and Sharing · See CAMS
custom software · 22
customer · 59, 60, 61
CY · 175
cycle time · 162
Cyclic Redundancy Check · See CRC

D

damage · 80
data analysis tools · 3
data leakage · 3
data loss · 65
debt · 162
declarative programming · 162
defect · 167
defect tracking · 163
Definition of Done · See DoD
Definition of Ready · See DoR
Definitional Uncertainty · See DU
Definitive Media Library · See DML
Demming wheel · 166
deployment · 159
deployment pipeline · 161
deployment team · 130
design · 1, 4, 138, 139, 162, 173, 178
Dev engineer · 1
development · 159, 160, 163, 165, 166, 168, 170, 171, 172, 173
Development & Operations · See DevOps
development ritual · 163
Development Value System · See DVS
Development, Test, Acceptance and Production · See DTAP
deviation · 7, 27, 118
DevOps · III, IV, 1, 2, 3, 4, 5, 11, 12, 21, 24, 25, 26, 28, 29, 35, 43, 137, 138, 139, 140, 141, 142, 145, 146, 147, 148, 151, 155, 159, 161, 163, 169, 173, 175
- Lemniscate · 1, 2, 4, 35, 138, 140, 141
- team · 1, 24, 25, 26, 139, 140, 145
DevOps engineer · 159, 161, 162, 163, 170, 171, 172
DevOps team · 159, 160, 161, 162, 164, 168, 173
digitisation · 23, 76
disaster · 80
DML · 175
DNS · 176
DoD · 2, 130, 136, 146, 149, 165, 176
Domain Name System · See DNS
DoR · 176
downward spiral · 163
DTAP · 170, 176
DTAP environments · 170
DU · 176
DVS · 4, 10, 11, 12, 13, 14, 15, 33, 34, 36, 38, 40, 41, 42, 43, 125, 132, 137, 149, 176

E

E2E · 176
eavesdropping · 79, 81
eclipse · 80
e-mail pass around · 163
emerging design · 3, 139
End User eXperience Monitoring · See EUX
endpoint · 3
End-to-End · See E2E
Enterprise Resource Planning · See ERP
Enterprise Service Bus · See ESB
Entity Relation Diagram · See ERD
epic · 130, 131, 176
Epic Solution Approach · See ESA
epic user story · 131
equipment failure · 80
ERD · 176
ERP · 24, 176
error path · 163
ESA · 176
ESB · 176
E-shaped · 164
ETL · 176
EUX · 176
event · 168
event register · 45
evidence · 9, 13, 18, 22, 23, 26, 27, 28, 33, 34, 36, 38, 42, 45, 46, 57, 69, 82, 84, 86, 87, 90, 91, 107, 110, 111, 112, 117, 118, 133, 142, 148, 149, 150
evidence collector · 46, 148
evidence criteria · 45
evolutionary project management · 128
eXtensible Markup Language · See XML
external audit · 59, 109, 114, 122, 123, 142
external auditor · 23
external issue · 14, 45, 58, 74
Extract Transform & Load · See ETL
eXtreme Programming · See XP

F

failure · 160
failure of communication links · 79
FAT · 159, 176
feature · 131, 163, 164, 173
feature driven development · 128
Feature Solution Approach · See FSA
feature toggle · 163
feedback · 162, 163, 166, 169, 172
feedforward · 163
finding criteria · 45, 84, 86
fire · 79
flood · 80

flow · 162, 165, 166, 167, 168, 169, 170, 172, 173
framework · 168
framework of standards · 46
fraud · 79, 81
FSA · 176
Functional Acceptance Test · See FAT
functionality · 2, 3, 17, 18, 88, 89, 112, 127, 128, 129, 131, 146

G

Gaussian distribution · 159, 163
GCC · 176
GDPR · 8, 14, 22, 76, 176
Gene Kim · 163, 168, 172
General Computer Controls · See GCC
General Data Protection Regulation · See GDPR
Generic & Specific Acceptatiecriteria · See GSA
Gherkin language · 32, 33, 111
GIT · 176
Given When Then · 164, See GWT
Global Information Tracker · See GIT
governance · 4, 11, 12, 25, 26, 33, 40, 49, 53, 55, 69, 109
Graphical User Interface · See GUI
green field · 164
growth model · 22, 32
GSA · 176
GUI · 176
GWT · 164, 176

H

Hand-off Readiness Review · See HRR
happy path · 159, 160, 164
hardware · 162, 165, 173
hide user identity · 79
holistic approach · 17
holocracy · 164
horizontal splitting of feature · 164, 173
HRM · 2, 28, 29, 31, 176
HRR · 176
Human Resource Management · See HRM

I

IaC · 159, 165, 176
ICT · 165, 176
ID · 176
ideal test pyramid · 172
idempodent · 164
IDentifier · See ID
impact rate · 74
impact severity · 73, 74, 77, 78
impact type · 73, 74, 77
imparative programming · 164
impediment · 131
incident criteria · 45
incrementally · 3, 33, 127
Independent, Negotiable, Valuable, Estimatable, Small and Testable · See INVEST
information asset · 4, 62, 63, 67, 72, 91, 92
Information assets, People, Organisation, Products and services, Systems and processes · See IPOPS
Information Communication Technology · See ICT
information management · 12, 137
information radiator · 165
information security
- asset · 61, 91
- auditing engine · 46, 111, 149
- incident · 49, 50, 52, 70, 75, 84, 88, 109, 119, 122, 123, 149
- policy · 4, 45, 49, 51, 55, 56, 57, 64, 65, 66, 72, 86, 97, 105, 114, 118
- risk · 2, 34, 42, 49, 51, 82, 83, 86, 91, 92, 93, 94, 97, 99, 116, 150
Information Security Management System · See ISMS
Information Security Value System · See ISVS
Information Standardisation Organisation · See ISO
Information Technology · See IT
Information Technology Infrastructure Library · See ITIL
Information Technology Service Management · See ITSM
Infosec · 165
Infrastructure as Code · See IaC
infrastructure component · 165
infrastructure management · 165
Infrastructure Risk · See IR
integrity · 2, 14, 64, 87, 88, 89, 90
interested parties register · 45, 63
interested party · 4, 13, 14, 45, 49, 50, 58, 59, 61, 62, 65, 66, 85, 107, 109, 111, 112, 116, 125, 133, 137
internal audit · 46, 52, 113, 114, 118, 119
internal audit result · 116
internal audit time · 116
internal issue · 14, 71, 72, 74, 76, 94, 96
internal issue register · 72
International Standard On Assurance Engagements · See ISAE
intruder · 64
intrusion · 3, 38
INVEST · 164, 176
IP address · 165
IPOPS · 45, 69, 70, 71, 72, 73, 74, 176
IR · 176
ISAE · 176
I-shaped · 164
ISMS · 176
ISO · 176

ISO 27001 · 4, 7, 8, 10, 13, 23, 28, 38, 39, 42, 46, 56, 58, 61, 62, 67, 72, 76, 86, 94, 101, 103, 104, 109, 111, 114, 115, 117, 118, 121, 136, 142
issue · 4
issue register · 45
IST · 166
IST – SOLL – Migration path modelling · 7
ISVS · 4, 7, 10, 11, 12, 13, 14, 15, 19, 32, 33, 34, 36, 38, 39, 40, 41, 42, 43, 45, 46, 48, 49, 50, 51, 52, 53, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 69, 70, 71, 72, 73, 74, 75, 76, 78, 79, 82, 84, 91, 92, 93, 94, 95, 96, 97, 99, 101, 104, 105, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 125, 132, 136, 137, 140, 142, 176
IT · 163, 167, 172, 176
iteratively · 3, 33
ITIL · 4, 10, 11, 12, 41, 138, 176
ITSM · 167, 176

J

Java Virtual Machine · See JVM
Ji-Kotei-Kanketsu · See, See JKK
JIT · 165, 166, 176
JKK · 165, 176
job description · 56
Just In Time · See JIT
JVM · 176

K

Kaizen · 165, 167
Kaizen Blitz (or Improvement Blitz) · 166
Kaizen in advance · 166
Kanban · 12, 166, 167, 173
Key Performance Indicator · See KPI
kibana dashboard · 166
KPI · 55, 63, 64, 65, 67, 150, 166, 167, 173, 176

L

LAN · 176
latent defect · 166
Launch Readiness Review · See LRR
launching guidance · 166
laws and regulations · 17, 18, 23, 57, 76, 139
LCM · 176
LDAP · 176
lead monitor · 7, 36
Lead Time · 166, See LT
leaked information · 79
Lean · 172, 173
Lean software development · 128
Lean tool · 166
learning culture · 167
lifecycle · 163, 165
LifeCycle Management · See LCM
lightning flash · 79
Lightweight Directory Access Protocol · See LDAP
Local Area Network · See LAN
log · 168
logging level · 167
loosely coupled architecture · 167
loosely coupled services · 167
loss of electricity · 80
LRR · 166, 177
LT · 10, 56, 58, 61, 63, 65, 70, 74, 75, 77, 78, 83, 91, 94, 97, 99, 101, 104, 105, 110, 113, 115, 117, 118, 119, 121, 122, 166, 177

M

maintenance error · 79
malicious code · 79
managed object · 46
Management Information · See MI
manual testing · 148
manufacturing process · 173
marker · 49
MASR · 46, 100, 103, 177
maturity · 5, 19, 24, 25, 26, 145, 147, 148, 150
Mean Time Between Failure · See MTBF
Mean Time Between System Incidents · See MTBSI
Mean Time To Repair · See MTTR
measurement data · 7
measurement instruction · 27, 28
meta-data · 160
MFA · 64, 177
MI · 177
Michael Porter · 9, 10, 11
microservice · 167
microservice architecture · 167
Microsoft Operations Framework · See MOF
mini pipeline · 167
Minimal Viable Product · See MVP
Minimum Required Information · See MRI
Modify, Avoid, Share, Retain · *See* MASR
Module Test · See MT
MOF · 12, 177
monitor facility · 111
monitor function · 18, 27
monitoring · 168
monolithic · 168
MRI · 167, 177
MT · 177
MT member · 59, 60
MTBF · 177
MTBSI · 177
MTTR · 168, 177
muda · 168
Multi Factor Authentication · See MFA

MVP · 177

N

national bank · 59, 60
NC · 46, 49, 50, 52, 70, 71, 82, 84, 86, 87, 90, 121, 122, 123, 124, 177
NFR · 12, 161, 168, 177
Non Conformity · See NC
Non Functional Requirement · See NFR
non-compliance · 2

O

OAWOW · 177
obeya · 168
object code · 160
OLA · 177
One Agile Way of Working · See OAWOW
one piece flow · 168
operational best practice · 4, 69
Operational Level Agreement · See OLA
operations · 159, 163, 168, 170, 173
operations story · 168
Ops engineer · 1
Ops liaison · 168
organisation archetype · 168
organisational typology model · 168
outcome · 7, 8, 13, 14, 17, 26, 27, 32, 33, 36, 38, 63, 69, 71, 72, 73, 74, 75, 76, 77, 79, 83, 84, 85, 86, 94, 139, 140
over-the-shoulder · 169

P

PAAS · 177
package · 169
pair programming · 161, 169
password · 80, 87, 88, 89
PAT · 159, 177
pattern · 159, 169
PBI · 177
PDCA · 166, 167, 177
peer review · 169
peer to peer programming · 161
People, Process & Technology · See PPT
performance · 161, 167, 173, 177
Performance StressTest · See PST
PESTLE · 45, 69, 75, 76, 77, 177
PESTLE classification · 77
pipeline · 159, 165, 167, 170, 171, 173
Plan Do Check Act · See PDCA
Platform As A Service · See PAAS
Political, Economic, Sociological, Technological, Legislative, Environmental · See PESTLE
pollution · 80
POR · 177
post mortem · 169
power · 4, 23, 24
PPT · 31, 177
priority criteria · 45
privacy authority · 59, 60
Processing Time · See PT
product
- backlog · 15, 23, 25, 125, 128, 129, 131, 132, 136, 137, 139, 142, 145, 149, 169
- backlog item · 165
- owner · 59, 60, 90, 128, 129, 130, 131, 132, 169
- portfolio · 45, 63, 91, 92

Product Backlog Item · See PBI
Production AcceptatieTest · See PAT
production environment · 12, 32, 52, 111, 138, 167
programming paradigm · 169
Project or Organisational Risk · See POR
projectteam · 131
PSQL · 162
PST · 177
PT · 56, 58, 61, 63, 65, 70, 74, 75, 77, 78, 83, 91, 94, 97, 99, 102, 104, 105, 110, 113, 115, 117, 118, 119, 121, 122, 177
pull request process · 169

Q

QA · 169, 177
QA department · 26
QC · 177
quality · 1, 2, 3, 4, 14, 17, 19, 25, 26, 28, 32, 33, 38, 40, 53, 57, 58, 79, 109, 127, 130, 131, 136, 137, 139, 146, 149
Quality Assurance · See QA
Quality Control · See QC

R

RACI · 177
RASCI · 25, 26, 31, 177
RBAC · 177
Real User Monitoring · See RUM
recursion · 10
reduce batch size · 169
reduce number of handoffs · 169
refactoring · 19, 139
regulatory obligation · 2
release · 1, 138, 169
release manager · 169
release pattern · 169
repository · 160, 161, 169, 170
REpresentational State Transfer Application Programming Interface · See REST API
requirement · 160, 166, 168, 171, 177
residual risk · 46
Responsibility, Accountable, Consulted and Informed · See RACI
Responsibility, Accountable, Supporting, Consulted and Informed · See RASCI
REST-API · 46, 149, 177

restrisico · 106
retrospective · 163
Return On Investment · See ROI
review · 163
risico · 160, 166
  - register · 99
risk
  - assessment · 4, 15, 45, 51, 69, 82, 83, 96, 97, 98, 103, 104, 134, 135, 136
  - control · 46, 149, 150
  - evaluation · 97
  - identification · 84, 95, 96, 98
  - management · 8, 13, 17, 18, 27, 34, 40, 71, 75, 84, 104
  - register · 46
  - treatment · 46, 51, 103, 134, 135
  - treatment option · 15, 51, 69, 83, 99, 100, 101, 102, 103, 104
  - treatment plan · 15, 97, 101, 105, 106, 107, 108
roadmap · 21, 25, 32, 33, 105, 106, 107, 108, 120, 139, 140, 145
ROI · 177
Role-based access control · See RBAC
rootcause analyse · 167
RUM · 177

S

SA · 177
sad path · 169
SAFe · 177
SAFe framework · 25
safety check · 169
Sarbanes Oxley · See SoX
SAT · 177
SBAR · 170, 177
SBB · 48, 177
SBB-A · 177
SBB-I · 177
SBB-T · 48, 177
SBL · 133, 135
Scaled Agile Framework · See SAFe
S-CI · 177
scope · 4, 8, 14, 42, 45, 49, 50, 55, 61, 62, 63, 64, 65, 66, 69, 86, 91, 93, 97, 106, 108, 109, 113, 114, 116, 125, 140
Scrum master · 24, 128, 130, 131
Scrum team · 130, 131
Secure Sockets Layer · See SSL
security · 2, 161, 168, 169, 170, 172
  - analist · 70, 71, 75, 79, 83, 92, 94, 97, 99, 100, 102, 104, 105, 106, 110, 111, 119, 120, 121, 122, 123
  - control · 34
  - goal · 7, 25, 39, 49, 143
  - manager · 50
  - officer · 50, 51, 52
  - policy · 39
Security Acceptance Test · See SAT
security officer · 161
security practice · 4, 39, 40, 45, 53, 55, 57, 58, 61, 63, 65, 69, 109, 123, 125, 132, 140, 141
security practices · 122
self service capability · 170
service · 177
Service Level Agreement · See SLA
service portfolio · 45, 63, 91, 92
Service Value System · See SVS
shared goals · 170
Shareholder · 59, 60
shift-left organisation · 21
silo · 173
Simian army · 170, 172
Simple Network Management Protocol · See SNMP
SIT · 178
Situation, Background, Assessment, Recommendation · See SBAR
skills · 164
SLA · 178
SLA norms · 23, 49
slow feedback · 18
SM · 178
SMART · 45, 63, 64, 167, 178
SME · 178
smoke testing · 170
SNMP · 178
SoA · 46, 178
social engineering · 80
SoE · 2, 3, 24, 171, 178
software · 160, 171, 173
Software Configuration Item · See S-CI
SoI · 2, 3, 171, 178
SOLL · 166
SoR · 2, 3, 24, 171, 178
sourcecode · 160, 161, 163, 170, 171, 172
SoX · 178
Specific, Measurable, Accountable, Realistic, Timely · See SMART
Spotify · 25
sprint · 3, 129, 132, 135, 136, 163
  - backlog · 131
sprint execution · 163
sprint planning · 163
SQL · 178
SRC · 45, 46, 55, 56, 57, 59, 60, 61, 62, 63, 64, 65, 66, 90, 97, 106, 108, 113, 114, 118, 119, 120, 122, 123
SRC-board · 46, 56, 57, 60, 106
SRC-board member · 59
SRG · 178
SSL · 178
ST · 178
stakeholder · 2, 9, 14, 17, 31, 49, 50, 55, 58, 59, 66, 84, 130, 131, 164
standard deviation · 170
standard operations · 170
Standard Rules & Guidelines · See SRG
stand-up · 163
Statement of Applicability · See SoA
static analysis · 171
Strategic IS Architecture · See SA

Strategic Match · See SM
strike · 80
Structured Query Language · See SQL
Subject Matter Expert · See SME
supplier · 59
sustainable · 57, 76
SVS · 4, 7, 10, 11, 12, 13, 14, 15, 33, 34, 36, 38, 40, 41, 42, 43, 69, 70, 71, 96, 106, 125, 137, 178
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 · 45
System context diagram · 63, 145
system development · 2, 4, 10, 125, 126, 127, 128, 138
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 · 159
target · 1, 7, 8, 17, 18, 19, 22, 26, 33, 37, 49, 65, 67
Target Operating Model · See TOM
task · 165
TCO · 178
TCP · 178
TDD · 171, 178
Team Foundation Server · See TFS
technical debt · 21, 25, 140, 148, 162, 163
Technical Information Security Officer · See TISO
Technical Uncertainty · See TU
technology adaption curve · 171
technology executive · 171
template · 48, 57, 59, 76, 77, 84, 85, 86, 87, 88, 90, 91, 92, 94, 95, 96, 97, 98, 100, 101, 102, 105, 107, 111, 115, 116, 121, 122, 123, 135, 136, 142
terrorist attack · 80
test
- case · 159, 160, 161
- criteria · 83, 84, 96
- harness · 171
Test Driven Development · See TDD
tester · 163
TFS · 178
The Agile Manifesto · 171
the ideal testing automation pyramid · 172
The Lean movement · 172
the non-ideal testing automation inverted pyramid · 172
The Three Ways · 168, 172
theft · 79
theory of constraints · 172
threat · 79, 81
Time To Market · See TTM
TISO · 91, 178
TOM · 7, 8, 11, 27, 33, 34, 42, 178
tool-assisted code review · 172
Total Cost of Ownership · See TCO
Toyota Kata · 173
Toyota Production System · See TPS
TPS · 9, 178
traceability · 18, 22, 23, 120, 142, 146
transformation team · 173
Transmission Control Protocol · See TCP
transport · 64
trigger criteria · 84
trunk · 170
T-shaped · 164
TSQL · 162
TTM · 178
TU · 178

U

UAT · 178
UML · 178
Unified Modeling Language · See UML
Unit Test · See UT
upfront design · 3
use case · 4, 18, 45, 46, 47, 48, 50, 51, 52, 56, 57, 58, 61, 63, 64, 65, 66, 67, 70, 71, 74, 75, 78, 79, 82, 84, 85, 86, 91, 92, 93, 94, 97, 99, 100, 101, 104, 105, 109, 111, 113, 114, 115, 116, 117, 118, 119, 121, 122, 123, 134, 141, 145
use case diagram · 4, 45, 46, 47, 48, 134, 141, 145
User Acceptance Test · See UAT
user error · 79
User eXperience Design · See UX design
user story · 2, 3, 131
UT · 178
UX design · 178

V

value chain · 4, 9, 10, 11, 12, 17, 26, 31, 36, 38, 39, 40, 45, 48, 49, 53, 55, 61, 62, 63, 74, 77, 140, 145
value stream · 2, 4, 7, 8, 9, 10, 11, 12, 13, 18, 19, 22, 26, 32, 37, 38, 39, 41, 45, 46, 48, 49, 50, 53, 62, 63, 71, 74, 77, 84, 85, 86, 87, 88, 89, 124, 132, 138, 139, 140, 145, 146, 147, 149, 150, 163, 167, 170, 171, 173, 178
value stream manager · 50
Value Stream Mapping · See VSM
value system · 11, 12, 13, 33, 36, 37, 38, 42, 43, 150
vandalism · 80
velocity · 132, 160
vertical splitting of feature · 173

violation of law · 80
virtualised environment · 173
Vision · 4, 22
visualisatie · 173
Voice over Internet Protocol · See VOIP
VOIP · 178
VSM · 173, 178
vulnerable · 79, 80, 85

W

walking skeleton · 173
WAN · 178
war room · 168
waste · 160, 162, 165, 166, 168, 172, 173
waste reductie · 173
waterfall project · 2
Way of Working · See WoW
Westrum · 168, 169
wet- en regelgeving · 18
Wide Area Network · See WAN
Windows Management Instrumentation · See WMI
WIP · 178
WMI · 178
Work In Progress · See WIP
workflow · 162
WoW · 178

X

XML · 178
XP · 12, 128, 178

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

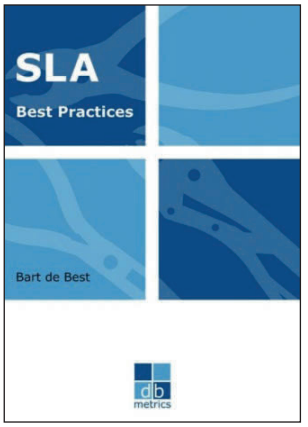
De eerste stap van een leven lang leren.

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

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

Auteur	: Bart de Best
Uitgever	: Leonon Media, 2011
ISBN (NL)	: 978 90 7150 1456

Cloud SLA
The best practices of cloud service level agreements

More and more organisations are opting to replace traditional ICT services with cloud services. Drawing up effective SLAs for traditional ICT services is a real challenge for many organisations. With the advent of cloud services, this initially seems much simpler, but soon the difficult questions such as data ownership, information links and security are addressed. This book describes what cloud services are. The risks that organisations run when entering into contracts and SLAs are discussed. Based on a long list of risks and countermeasures, this book also provides recommendations for the design and content of the various service level management documents for cloud services.
This book first defines the term 'cloud' and then describes various aspects such as cloud patterns and the role of a cloud broker. The core of the book concerns the discussion of contract aspects, service documents, service designs, risks, SLAs, and cloud governance. To enable the reader to immediately get started with cloud SLAs, the book also includes checklists of the following documents: Underpinning Contract (UC), Service Level Agreement (SLA), File Financial Agreements (DFA), Dossier Agreements and Procedures (DAP), External Spec Sheets (ESS) and Internal Spec Sheets (ISS).

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

SLA Templates
A complete set of SLA templates

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

The following templates are included in this book:

- Service Level Agreement (SLA)
- Underpinning Contract (UC)
- Operational Level Agreement (OLA)
- Document Agreement and Procedures (DAP)
- 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 stringentere wet- en regelgeving, de hogere eisen die gesteld worden vanuit regievoering over uitbestede services en de toegenomen complexiteit van informatiesystemen. Om op de gewenste servicenormen te kunnen sturen, is het belangrijk om een Performance Measurement System (PMS) te ontwikkelen. Daarmee kunnen niet alleen de te leveren ICT-services worden gemeten, maar tevens de benodigde 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.

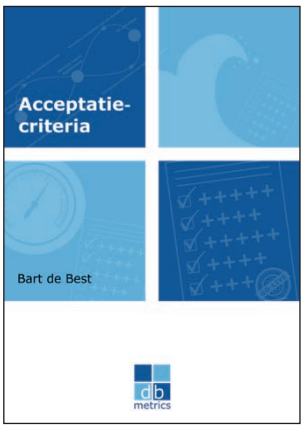
Auteur	: Bart de Best
Uitgever	: Leonon Media, 2011
ISBN (NL)	: 978 90 7150 1470

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

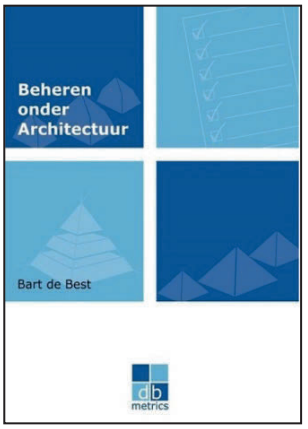


Acceptatiecriteria

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

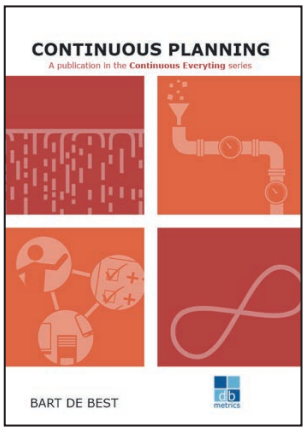
DevOps Architecture Best Practices

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

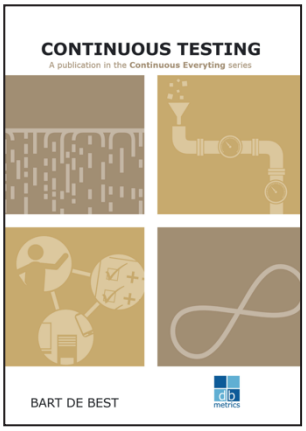
Continuous Design

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

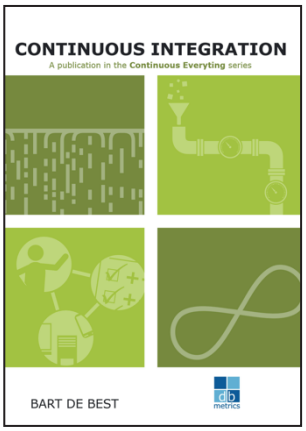


Continuous Testing

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

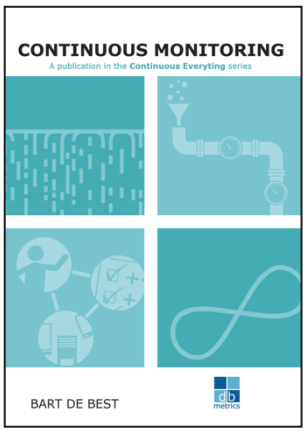
Continuous Deployment

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 are observed in production of the CI/CD secure pipeline, making recovery actions faster and cheaper, leading to a waste reduction.

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

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



Continuous Monitoring

A publication in the Continuous Everything series.

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

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

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

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

Continuous Learning

A publication in the Continuous Everything series.

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

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

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

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

Continuous Assessment

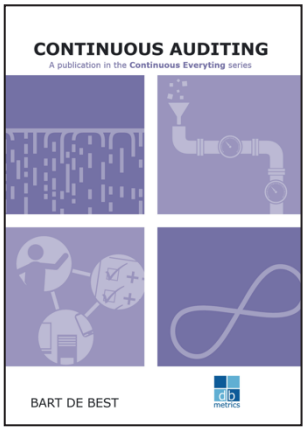
A publication in the Continuous Everything series.

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

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

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

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



Continuous Auditing

A publication in the Continuous Everything series.

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

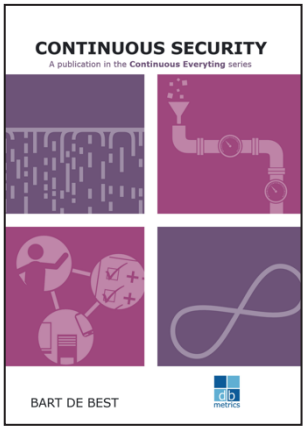
As a result, compliance risks are prevented by already thinking about which risks to mitigate or eliminate from the requirements and the design based on them.

This book is a publication in the continuous everything series.

The content consists of an explanation of the continuous auditing pyramid model that describes the six steps to give substance to continuous auditing, namely: determining scope, determining goals, identifying risks, realising controls, setting up monitoring facilities and demonstrating effectiveness of controls.

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

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



Continuous Security

A publication in the Continuous Everything series.

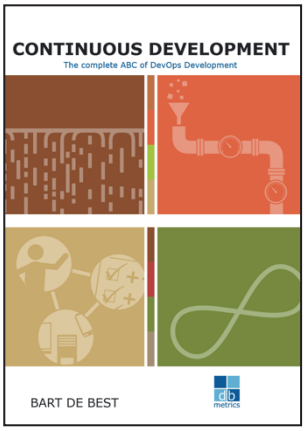
Continuous security is an approach that aims to keep an organisation in control from three perspectives:

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

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

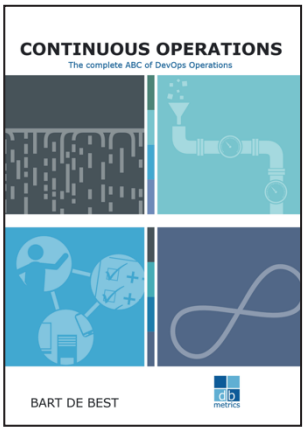
A publication in the Continuous Everything series.

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

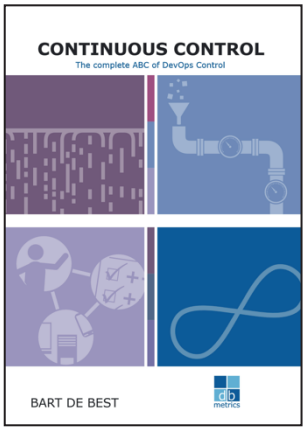
A publication in the Continuous Everything series.

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

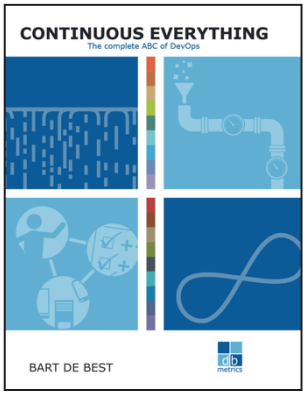
A publication in the Continuous Everything series.

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

Author	: Bart de Best
Publisher	: Leonon Media, 2022
ISBN (NL)	: 978 94 91480 195
ISBN (UK)	: 978 94 91480 201



Continuous Everything

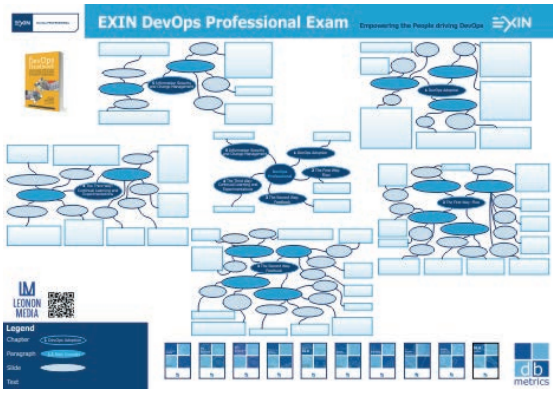
A publication in the Continuous Everything series.

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



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

DevOps Poster

DevOps Professional Exam Poster

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

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

CONTINUOUS SECURITY

A publication in the
Continuous Everyting
series

Bart de Best

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

1. **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.**
2. **The development perspective: Development value streams are in control by integrally including the non-functional require-ments for information security in the development.**
3. **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 requirements.**

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.

db metrics

ISBN 978-9-491480-18-8