

DevOps Development Best Practices

A set of best practices
for the development of software
in a DevOps environment

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
Manuscripts / Authors : redactie@leonon.nl

© 2021 Leonon Media

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

Title : DevOps Development
Subtitle : A set of best practices for the development of software in a DevOps environment
Date : 10 January 2021
Author : Bart de Best
Publisher : Leonon Media
ISBN13 : 978 94 92618 184
Edition : First edition, 10 January 2021

©2021, Leonon Media

Nothing in this publication may be reproduced and/or made public by print, photocopy, microfilm, or any other means without the prior written permission of the publisher.

TRADEMARK NOTICES

ArchiMate® and TOGAF® are registered trademarks of The Open Group.

COBIT® is a registered trademark of the Information Systems Audit and Control Association (ISACA) / IT Governance Institute (ITGI).

ITIL® and PRINCE2® are registered trademarks of Axelos Limited.

***"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.....	3
1.4.1	CHAPTER 2: CONTINUOUS PLANNING	3
1.4.2	CHAPTER 3: CONTINUOUS DOCUMENTATION.....	3
1.4.3	CHAPTER 4: CONTINUOUS TESTING	3
1.4.4	CHAPTER 5: CONTINUOUS INTEGRATION	3
1.4.5	CHAPTER 6: OTHER SUBJECTS	3
1.5	APPENDICES.....	4
1.6	APPROACH	4
1.7	READING GUIDELINE	4
2	CONTINUOUS PLANNING.....	5
2.1	INTRODUCTION.....	5
2.2	ROADMAP TO VALUE	5
2.2.1	DEFINITION	5
2.2.2	RISKS.....	5
2.2.3	ANTI-PATTERN.....	5
2.2.4	PATTERN.....	5
2.2.5	EXAMPLE	5
2.2.6	IMPLEMENTATION	6
3	CONTINUOUS DOCUMENTATION	9
3.1	INTRODUCTION.....	9
3.2	AGILE DESIGN PYRAMID	10
3.2.1	DEFINITION	10
3.2.2	RISKS.....	10
3.2.3	ANTI-PATTERN.....	10
3.2.4	PATTERN.....	10
3.2.5	EXAMPLE	10
3.3	CLEANCODE	12
3.3.1	DEFINITION	12
3.3.2	RISKS.....	12
3.3.3	ANTI-PATTERN.....	12
3.3.4	PATTERN.....	12
3.3.5	EXAMPLE	12
3.4	ANNOTATION	13
3.4.1	DEFINITION	13
3.4.2	RISKS.....	13
3.4.3	ANTI-PATTERN.....	13
3.4.4	PATTERN.....	13
3.4.5	EXAMPLE	13
3.5	DOCUMENTATION COVERAGE	14
3.5.1	DEFINITION	14
3.5.2	RISKS.....	14
3.5.3	ANTI-PATTERN.....	14

3.5.4	PATTERN	14
3.5.5	EXAMPLE	14
3.6	REPOSITORY DRIVEN DOCUMENTATION	14
3.6.1	DEFINITION	14
3.6.2	RISKS	14
3.6.3	ANTI-PATTERN	14
3.6.4	PATTERN	14
3.6.5	EXAMPLE	14
3.7	RISK BASED DOCUMENTATION	15
3.7.1	DEFINITION	15
3.7.2	RISKS	15
3.7.3	ANTI-PATTERN	15
3.7.4	PATTERN	15
3.7.5	EXAMPLE	15
4	CONTINUOUS TESTING	17
4.1	INTRODUCTION	17
4.1.1	FAST FEEDBACK	17
4.1.2	SLOW FEEDBACK	18
4.2	BEHAVIOR DRIVEN DEVELOPMENT MATRIX	18
4.2.1	DEFINITION	19
4.2.2	RISKS	19
4.2.3	ANTI-PATTERN	19
4.2.4	PATTERN	19
4.2.5	EXAMPLE	19
4.3	TEST DRIVEN DEVELOPMENT	20
4.3.1	DEFINITION	20
4.3.2	RISKS	20
4.3.3	ANTI-PATTERN	20
4.3.4	PATTERN	20
4.3.5	EXAMPLE	21
4.4	TEST TYPE-MATRIX	21
4.4.1	DEFINITION	21
4.4.2	RISKS	21
4.4.3	ANTI-PATTERN	21
4.4.4	PATTERN	22
4.4.5	EXAMPLE	22
4.5	TEST TECHNIQUE-MATRIX	22
4.5.1	DEFINITION	22
4.5.2	RISKS	22
4.5.3	ANTI-PATTERN	22
4.5.4	PATTERN	22
4.5.5	EXAMPLE	23
4.6	GENERIC TEST STRATEGY	23
4.6.1	DEFINITION	23
4.6.2	RISKS	23
4.6.3	ANTI-PATTERN	24
4.6.4	PATTERN	24
4.6.5	EXAMPLE	24
4.7	OTHER	24
4.7.1	CODE REVIEW	25
4.7.2	PAIR PROGRAMMING	25

4.7.3	TEST AUTOMATION	25
4.7.4	TEST PATTERN.....	25
5	CONTINUOUS INTEGRATION.....	27
5.1	INTRODUCTION.....	27
5.2	CENTRALE SOURCE CODE REPOSITORY.....	27
5.2.1	DEFINITION	27
5.2.2	RISKS.....	28
5.2.3	ANTI-PATTERN.....	28
5.2.4	PATTERN.....	28
5.2.5	EXAMPLE	28
5.2.6	IMPLEMENTATION	28
5.3	CENTRALE BINARY REPOSITORY.....	29
5.3.1	DEFINITION	29
5.3.2	RISKS.....	29
5.3.3	ANTI-PATTERN.....	29
5.3.4	PATTERN.....	29
5.3.5	EXAMPLE	29
5.3.6	IMPLEMENTATION	29
5.4	DOR AND DOD	29
5.4.1	DEFINITION	29
5.4.2	RISKS.....	30
5.4.3	ANTI-PATTERN.....	30
5.4.4	PATTERN.....	30
5.4.5	EXAMPLE DOR.....	30
5.4.6	EXAMPLE DOD.....	31
5.5	REFACTORING	32
5.5.1	DEFINITION	32
5.5.2	RISKS.....	32
5.5.3	ANTI-PATTERN.....	33
5.5.4	PATTERN.....	33
5.5.5	EXAMPLE	33
5.6	SHORT LIVE BRANCHES.....	33
5.6.1	DEFINITION	33
5.6.2	RISKS.....	33
5.6.3	ANTI-PATTERN.....	33
5.6.4	PATTERN.....	33
5.6.5	EXAMPLE	33
5.7	STANDARD, RULES & GUIDELINES.....	34
5.7.1	DEFINITION	34
5.7.2	RISKS.....	34
5.7.3	ANTI-PATTERN.....	34
5.7.4	PATTERN.....	34
5.7.5	EXAMPLE	34
5.8	TRACEABILITY.....	34
5.8.1	DEFINITION	34
5.8.2	RISKS.....	35
5.8.3	ANTI-PATTERN.....	35
5.8.4	PATTERN.....	35
5.8.5	EXAMPLE	36
5.9	WASTE REDUCTION	37
5.9.1	DEFINITION	37

5.9.2	RISKS	37
5.9.3	ANTI-PATTERN	37
5.9.4	PATTERN	37
5.9.5	EXAMPLE	38
6	OTHER TOPICS	39
6.1	REST API'S	39
6.1.1	DEFINITION	39
6.1.2	RISKS	39
6.1.3	ANTI-PATTERN	39
6.1.4	PATTERN	39
6.2	SECURITY IN THE PIPELINE	40
6.2.1	DEFINITION	40
6.2.2	RISKS	40
6.2.3	ANTI-PATTERN	40
6.2.4	PATTERN	40
	APPENDIX A, LITERATURE LIST	43
	APPENDIX B, GLOSSARY	45
	APPENDIX C, ABBREVIATIONS	51
	APPENDIX D, AGILE TOOLS	53
	APPENDIX E, WEBSITES	57
	APPENDIX F, INDEX	59

Figures

FIGURE 1-1, COHERENCE OF CONTINUOUS EVERYTHING CONCEPTS.....	2
FIGURE 1-2, COHERENCE OF CONTINUOUS EVERYTHING CONCEPTS.....	2
FIGURE 2-1, ROADMAP TO VALUE [LAYTON 2017].	6
FIGURE 3-1, CONTINUOUS DOCUMENTATION.....	9
FIGURE 3-2, AGILE DESIGN PYRAMID.....	10
FIGURE 3-3, FILLED IN IDEAL DESIGN PYRAMID.	11
FIGURE 3-4, IMAGE OF THE IDEAL DESIGN PYRAMID ON THE AGILE PLANNING.	11
FIGURE 4-1, IDEAL TESTING PYRAMID.....	17
FIGURE 4-2, NON-IDEAL DESIGN PYRAMID.....	18
FIGURE 5-1, COHERENCE OF CONTINUOUS INTEGRATION CONCEPTS.	27
FIGURE 5-2, TEMPLATE SYSTEM BUILDING BLOCKS – INFORMATION.	36
FIGURE 5-3, EXAMPLE OF AN SBB-I FOR A COFFEE SERVICE.....	36
FIGURE 6-1, REST API.....	39

Tables

TABLE 1-1, CONTINUOUS EVERYTHING ASPECTS.....	2
TABLE 1-2, APPENDICES.....	4
TABLE 2-1, ROADMAP.....	6
TABLE 4-1, GHERKIN KEYWORDS.	19
TABLE 4-2, GHERKIN FEATURE FILE EXAMPLE.	20
TABLE 4-3, PYTHON UNIT TEST TEMPLATE.	21
TABLE 4-4, PYTHON UNIT TEST EXAMPLE.	21
TABLE 4-5, TEST TYPE MATRIX TEMPLATE.....	22
TABLE 4-6, TEST TYPE MATRIX EXAMPLE.	22
TABLE 4-7, TEST TYPE-MATRIX TEMPLATE.	23
TABLE 4-8, TEST TECHNIQUE MATRIX EXAMPLE.....	23
TABLE 4-9, TEST STRATEGY TEMPLATE.....	24
TABLE 4-10, TEST STRATEGY EXAMPLE.	24
TABLE 5-1, DOR EXAMPLE.	31
TABLE 5-2, DOD EXAMPLE.	32

Appendices

APPENDIX A, LITERATURE LIST.....	43
APPENDIX B, GLOSSARY.....	45
APPENDIX C, ABBREVIATIONS.....	51
APPENDIX D, AGILE TOOLS.....	53
APPENDIX E, WEBSITES.....	57
APPENDIX F, INDEX.....	59

Preface

This book has been compiled based on my Development & Operations (DevOps) practical experience. There is no framework or method or methodology for DevOps. In order to give structure to the content of this book, the book 'DevOps Assessments' has therefore been used. It defines the concept of Continuous Everything. A maturity model is also linked to this. In this book, the aspects of Continuous Integration, Continuous Testing and Continuous Documentation of the Continuous Everything concept have been selected. Then I chose the most important topics from these three aspects of Continuous Everything. For convenience, I have added Continuous Planning.

This book thus has four core chapters in which the DevOps Development best practices are expressed. By naming the risks to be controlled per subject, the importance of applying the subject becomes clear. I also indicate per subject what is not the objective in the form of an anti-pattern.

This has resulted in a book that is not an A-to-Z book, but rather a reference work to come to the implementation of DevOps best practices. Together with the book DevOps Assessments, it is a practical tool for putting DevOps Development on the map and further developing it.

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 which I provide. These can be found at www.dbmetrics.nl.

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

- J.A.E. (Jane) ten Have APG AM
- Dr. L.J.G.T. (Louis) van Hemmen BitAll B.V.
- F.J. (Fred) Ros RE RA Auditdienst Rijk
- D (Dennis) Wit ING

I hope you enjoy reading this book and, above all, good luck in applying Agile Design within your own organization.

If you have any questions or comments, please do not hesitate to contact me. Much time has gone into making this book as complete and consistent as possible. Should you find any shortcomings, I would appreciate it if you let me know, so that these issues can be processed in the next edition.

1 Introduction

Reading guideline:

This chapter describes the purpose of this book (1.1), the intended audience (1.2), the background (1.3), the structure (1.4) and finally some tips for reading this book (1.5, 1.6 and 1.7).

1.1 Goal

The aim of this book is to give tips and tricks when writing software in a DevOps context.

1.2 Target audience

The target group of this book are all employees who are involved in building an information system using a DevOps approach. This includes architects, Dev engineers, Ops engineers, Product owners, Scrum masters, Agile coaches and representatives of the user organization. This book is of course also very suitable for line managers, process owners, process managers, et cetera who are involved in the realization of the information provision through an Agile design.

Finally, there is a target group that does not develop or manage, but that determines whether the information provision meets the required criteria. This target group includes quality employees and auditors. They can use this book to identify risks that need to be taken or managed.

1.3 Background

For many decades, information systems have been written worldwide in hundreds of programming languages. With the application of DevOps becoming prominent, the question arises what a programmer should do differently than before the current DevOps time. The answer to this question is also the basis for the content of this book. However, there is no unambiguous answer to this question because DevOps is not a method or methodology. There is no authentic source that defines what DevOps is. DevOps is the concept of storytelling. For most people, the image of DevOps is based on DevOps publications in which experts give their opinion on how to position DevOps.

In order to get a foundation for this book, the book DevOps Assessment [BEST 2019] has been used as a framework. The Continuous Everything aspects selected from this book are:

- Continuous Integration
- Continuous Testing
- Continuous Documentation
- Other subjects

The following characteristics are not included in this list:

- Continuous Monitoring
- Continuous Delivery / Continuous Deployment
- Continuous Learning.

These aspects fall within the scope of the book DevOps Operations. These aspects naturally have a close relationship with what falls within the scope of this book. For Continuous Learning, this aspect applies to both Dev and Ops, but it has been chosen to include this within the book DevOps Operations.

Figure 1-1 provides an overview of the phases that are completed in a DevOps cycle.

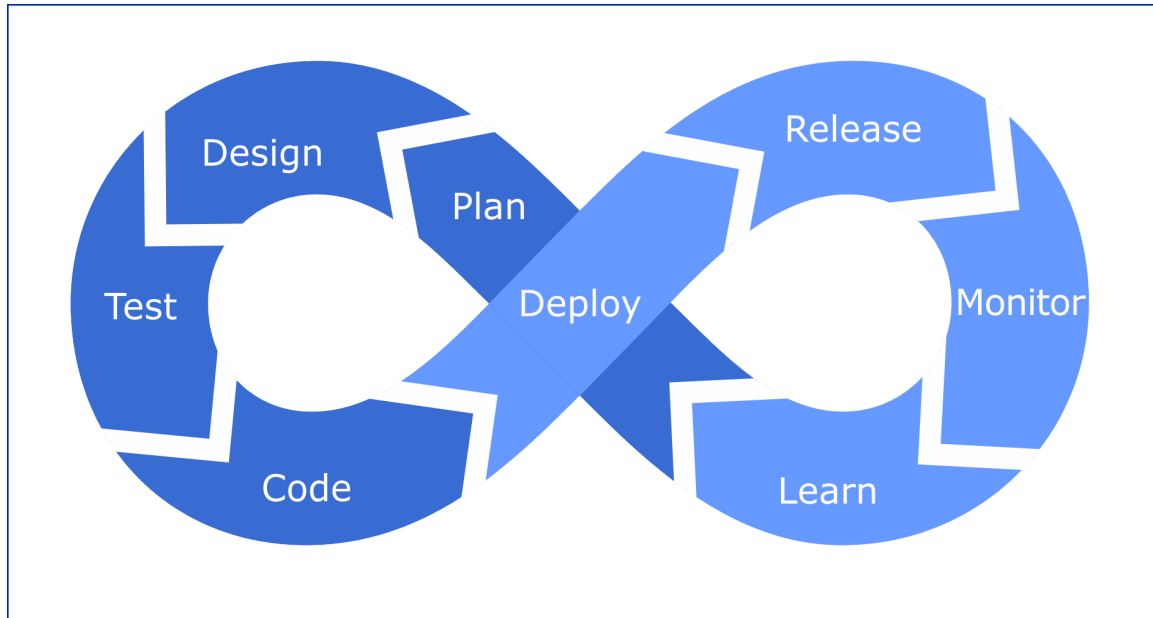


Figure 1-1, Coherence of Continuous Everything concepts.

Figure 1-1 shows an attempt to capture the phases of Continuous Everything in a sequence. However, this is not a correct representation of reality, because the phases overlap. This effect is shown in Figure 1-2. The numbers in this figure are defined in Table 1-1.

	1	2	3	4	5	6	7	8
1		V	V	V	V	V	V	V
2	V		V	V	V	V	V	V
3	-	V		V	V	V	V	-
4	-	-	V		V	V	V	-
5	-	-	-	-		-	V	-
6	-	-	-	-	-		V	-
7	V	V	V	V	V	V		V
8	V	V	V	V	V	V	v	

Figure 1-2, Coherence of Continuous Everything concepts.

Development		Operations	
1	Continuous Planning (Plan)	5	Continuous Deployment (Deploy)
2	Continuous Documentation (Design)	6	Continuous Deployment (Release)
3	Continuous Testing (Test)	7	Continuous Monitoring
4	Continuous Integration (Code)	8	Continuous Learning

Table 1-1, Continuous Everything aspects.

The Continuous Everything aspects included in Figure 1-2 should be read from left to right. For example, Continuous Testing (3) is used with Continuous Documentation (2) because the test cases fall under the heading of Continuous Documentation.

Appendix F, Index

A

acceptance criterion · 31, 32, 45, 46
 acceptance test · 45
 Agile
 - application documentation · 20
 - approach · 4
 - coach · 1
 - description · 9
 - design · 1, 14
 - design pyramid · 3
 - infrastructure · 45
 - method · 10, 19
 - project · 35
 - scaling model · 5
 alternate path · 45
 Amazon Web Services · See AWS
 annotatie · 3
 anti-pattern · 9, 17, 25, 45
 artefact · 45
 automated test · 45
 AWS · 51

B

backward traceability · 34
 bad path · 45
 baseline · 35
 BDD · 13, 17, 18, 45, 51
 Behavior Driven Development · See BDD
 BI · 51
 branch testing · 24
 branche · 3
 build · 46
 Business Intelligence · See BI
 business value · 32, 45

C

C/A · 12, 51
 capability · 46
 CD · 17,
 central binary repositories · 3
 central source code repositories · 3
 Chief Information Officer · See CIO
 Chief Technology Officer · See CTO
 CI · 17,
 CI / CD secure deployment pipeline · 27
 CI / CD secure pipeline · 17
 CIA · 51
 CIA rating · 15
 CIO · 51
 Cleancode · 3
 cloud · 45
 cloud configuration · 28

cloud configuration file · 45
 CMMI · 67
 code review · 3
 commit code · 45
 commit stage · 45
 Completeness / Accurateness · See C/A
 component · 46, 47, 48, 49
 Confidentiality, Integrity & Accessibility ·
 See CIA
 configuration management · 45
 container · 46
 continuïteit · 31
 Continuous
 - Delivery · 1
 - Deployment · 1, 2
 - Documentation · 1, 2, 3
 - Integration · 1, 2, 3
 - Learning · 1, 2
 - Monitoring · 1, 2
 - Planning · 2, 3
 - Testing · 1, 2, 3
 Continuous Delivery · See CD
 Continuous Integration · See CI
 control · 48
 Corporate DoD · 31
 CTO · 33, 51
 cycle time · 46

D

defect · 47
 Definition of Done · See DoD
 Definition of Ready · See DoR
 Definitive Media Library · See DML
 deployment · 45, 47
 deployment management · 35
 deployment pipeline · 45
 development · 45, 46, 49, 51
 Development & Operations · See DevOps
 development process · 20, 66
 DevOps · 51
 DevOps Operations · 1
 DevOps team · 67
 DML · 29, 51
 documentation coverage · 3
 DoD · 3, 27, 29, 31, 32, 51
 DoR · 3, 27, 51

E

Enterprise Service Buss · See ESB
 epic · 6
 error path · 46
 ESB · 39, 51
 estimable · 30
 event · 47
 everything as code · 9

F

fast feedback · 3
 FAT · 18, 22, 23, 24, 51
 feature · 46, 49
 feature file · 19
 feature toggle · 46
 feed forward · 46
 feedback · 46, 48
 flow · 47, 49
 Functional Acceptance Test · See FAT

G

GDPR · 31, 51
 General Data Protection Regulation · See
 GDPR
 Generic & Specific Acceptance criteria · See
 GSA
 Gherkin · 18, 20
 GIT · 14, 51
 Given When Then · 46, See GWT
 Global Information Tracker · See GIT
 governance · 67
 Graphical User Interface · See GUI
 GSA · 51
 GSA DoD · 31, 32
 GUI · 15, 51
 GWT · 18, 35, 51

H

happy path · 46

I

IaC · 45, 46, 47, 51
 ICT · 51
 ID · 51
 ideal testing pyramid · 48
 Identifier · See ID
 independent · 30
 Independent, Negotiable, Valuable,
 Estimable, Small and Testable · See
 INVEST
 Information Communication Technology ·
 See ICT
 information management · 67
 Information Security Management System ·
 See ISMS
 Information Security Value System · See
 ISVS
 Information Technology · See IT
 Information Technology Infrastructure
 Library · See ITIL
 Infrastructure as Code · See IaC
 infrastructure component · 47
 infrastructure management · 47
 INVEST · 30, 51

IP address · 46
 ISMS · 40, 51
 ISO 27001 · 31
 ISVS · 40, 51
 IT · 51
 ITIL · 51

J

Java Virtual Machine · See JVM
 Jira · 35
 JVM · 29, 51

K

Kaizen · 47
 Kanban · 5
 Key Performance Indicator · See KPI
 KPI · 47, 50, 51

L

latent defect · 47
 Lead Time · See LT
 Lean · 47, 49
 Lean tool · 47
 log · 47
 loosely coupled · 47
 LT · 12, 47, 51

M

machinecode · 29
 manufacturing process · 49
 markdown language · 9
 microservice · 47
 microservice architecture · 47
 Minimal Viable Product · See MVP
 Module Test · See MT
 monitoring · 47
 MT · 51
 MVP · 6, 51

N

negotiable · 30
 NFR · 47, 51
 NFR DoD · 31
 Non Functional Requirement · See NFR

O

operations · 5, 45, 47, 51

P

pair programming · 3, 25
 PAT · 22, 23, 24, 51
 patch · 29
 pattern · 45, 48
 PDCA · 47, 51
 peer review · 3
 PEN test · 26
 performance · 45
 Performance Stress Test · See PST
 pipeline · 45, 47, 50
 Plan Do Check Act · See PDCA
 Processing Time · See PT
 processor · 29
 product owner · 48
 Production Acceptance Test · See PAT
 programming paradigm · 48
 PST · 18, 22, 23, 24, 51
 PT · 12, 51
 pull request process · 48
 Python · 13, 21

Q

QA · 48, 51, 67
 QC · 51
 Quality Assurance · See QA
 Quality Control · See QC

R

refactoring · 12
 release · 2, 29, 32, 48
 release pattern · 48
 release planning · 5, 7
 repository · 45, 48
 repository driven documentation · 3
 REpresentational State Transfer Application
 Programming Interface · See REST API
 requirement · 19, 35, 45, 47, 51
 REST API · 3, 52
 risk · 66
 risk based documentation · 3
 roadmap · 5, 6
 Roadmap to Value · 3, 5
 root cause analyse · 47

S

Sarbanes Oxley · See SoX
 SAT · 18, 22, 23, 24, 52
 SBB · 52
 SBB-A · 52
 SBB-I · 20, 52
 SBB-T · 52
 Scrum · 5
 security · 47, 65
 Security Acceptance Test · See SAT

silo · 49
 SIT · 22, 23, 24, 52
 SLA · 65
 small · 30, 31
 SMART · 47, 52
 SoE · 48, 52
 SoI · 52
 SoR · 48, 52
 source code · 3, 13, 14, 35, 45, 48, 49
 SoX · 52
 spaghetti code · 12
 Specific, Measurable, Accountable,
 Realistic, Timely · See SMART
 SQL · 52
 SRG · 13, 32, 52
 ST · 52
 stakeholder · 46
 Standard Rules & Guidelines · See SRG
 state-of-the-art · 33
 Structured Query Language · See SQL
 System Building Block · See SBB
 System Building Block Application · See
 SBB-A
 System Building Block Infrastructure · See
 SBB-I
 System Building Block Technology · See
 SBB-T
 System Integration Test · See SIT
 System of Engagement · See SoE
 System of Records · See SoR
 system software · 28
 System Test · See ST
 Systems of Information · See SoI

T

TDD · 13, 17, 49, 52
 Team Foundation Server · See TFS
 test
 - able · 30, 31
 - automation · 3
 - case · 28, 35
 - data · 28
 - pattern · 3
 - result · 28
 - soort-matrix · 3
 - techniek-matrix · 3
 test case · 45
 Test Driven Development · See TDD
 test level · 22
 test technique matrix · 22
 test type matrix · 21
 TFS · 52
 The Agile Manifesto · 49
 the ideal testing automation pyramid · 48
 The Lean movement · 49
 the non-ideal testing automation inverted
 pyramid · 48
 tool · 55
 tool-assisted code review · 49
 top-down · 49
 traceability · 3

trunk · 48

U

UAT · 52

Unit Testing · See UT

unittest · 21

use case · 12, 35

use case diagram · 12

User Acceptance Test · See UAT

userstory · 31

UT · 52

V

valuable · 30, 31

value stream · 46, 47, 49, 52

Value Stream Mapping · See VSM

velocity · 45

versie · 29

virtualized environment · 49

vision statement · 5, 6

visualization · 49

VSM · 47, 49, 52

W

waste · 3, 45, 47, 49, 50

waste reduction · 3, 50

WIP · 50, 52

Work In Progress · See WIP

Epilogue

My experience is that the ideas that I capture in an article or a book keep evolving. In case you are going to work on a specific topic from this book in your own DevOps organization, I advise you to contact me. There may be additional articles or experiences in this area that I can share with you. This also applies inversely. If you have certain experiences that complement what is described in this book, I invite you to share this with me. You can reach me via my e-mail address bartb@dbmetrics.nl.

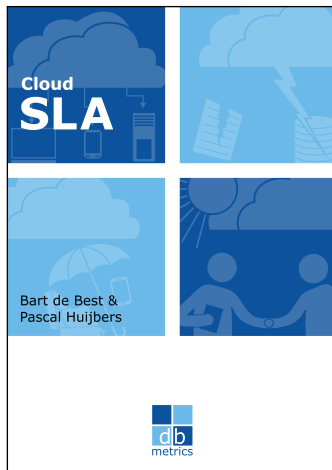
About the author



Drs. Ing. B. de Best RI. has been active in ICT since 1985. He worked primarily with the top 100 of Dutch business and government organizations. He has acquired experience in different roles within all aspects of system development, including operations for 12 years. After that, he focused on the subject of service management.

Currently, as a consultant, he is active in all aspects of the knowledge management cycle of service management, such as training ICT managers and service managers, advising service management organizations, improving service management processes and outsourcing (parts of) service management organizations. He graduated at both the HTS and University level in the management field.

Other books by this author



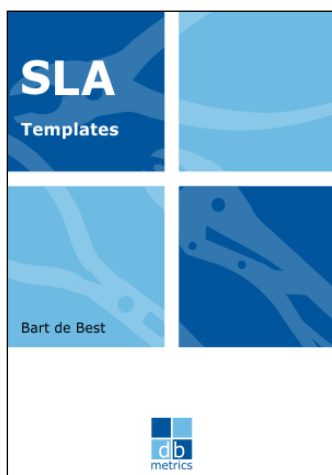
Cloud SLA

The best practices of cloud service level agreements

More and more organizations choose to replace traditional ICT services by cloud services. Setting up effective SLAs for traditional ICT services is a real challenge for many organizations. With the arrival of cloud services, this seems to be much simpler at first, but soon the hard questions come up like data ownership, information links and security. This book describes what cloud services are. The risks involved in 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 cloud and then describes various aspects like cloud patterns and the role of a cloud broker. The core of

the book is the discussion of contract aspects, service documents, service design, risks, SLAs and cloud governance. In order to allow readers to get started with Cloud SLAs, the book also includes checklists of the following documents: Underpinning Contract (UC), Service Level Agreement (SLA), Document Financial Agreements (DFA), Document Agreement and Procedures (DAP), External Spec Sheets (ESS) and Internal Spec Sheets (ISS).

Author : Bart de Best
 Publisher : Leonon Media, 2017
 ISBN (UK) : 978 94 92618 009
 ISBN (NL) : 978 90 71501 739



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



Agile Service Management with Scrum

On the way to a healthy balance between the dynamics of developing and the stability of managing the information provision

Using Agile software development is taking off. The terms Scrum and Kanban are already common to many organizations. Agile software development needs different requirements for the management of software. Many organizations are mastering this new challenge. In particular, the interaction between the Scrum development process and the support of the software that the Scrum development process has produced, is an important aspect. This book specifically discusses this interaction. Examples of topics that are discussed here are the service portfolio, SLAs and the handling of incident and change requests.

This book first defines the risk areas when implementing Scrum and Kanban. Next the various Agile terms and concepts are discussed. The content of Agile service management is described both at the organizational- as the process level. The relevant risks are specified for each of the service management processes. In addition, the implementation of each process within the context of Scrum is indicated.

Auteur : Bart de Best
 Uitgever : Leonon Media, 2015
 ISBN (NL) : 978 90 7150 1807
 ISBN (UK) : 978 94 9261 8085



Agile Service Management with Scrum Researched

On the way to a healthy balance between the dynamics of developing and the stability of managing the information provisions

Many companies are starting to apply Agile software development using Scrum or Kanban or have already implemented the new development process. Sooner or later the question arises how this development process relates to the service management processes. The book 'Agile Service Management with Scrum' has already addressed this interface and a number of risks per service management process have been identified. Countermeasures that can be taken are also defined. In a research at ten organizations these risks were presented, and they were asked how they deal with these risks. The research included the investigation into which Agile aspects are applied and in particular those of

Scrum or Kanban. Finally, each organization has carried out a maturity assessment for both the Agile development process and the change management process.

This book is the report about the research of the collaboration of Agile software development and service management processes in practice. The target group of this book includes all parties involved in the application of Agile software development and who would like to know how colleagues have shaped this crucial interface for successful service provision. In this book a short description is given of each organization about how the Agile development process has been designed.

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



DevOps Best Practices

Best Practices for DevOps

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

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

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



DevOps Assessments

A handy tool for DevOps teams to improve their skills

DevOps teams organise themselves and improve the Development (Dev) and Operations (Ops) aspects that are still underexposed. On the other hand, DevOps teams share knowledge and skills with other DevOps teams so that they reinforce each other.

This book provides a tool to make the DevOps teams aware of where they stand in terms of maturity and the next steps they can take to develop. To this end, this book offers two assessments. The first assessment is the DevOps Cube assessment based on 'The Three Ways' by Gene Kim. The second assessment is based on Continuous Everything whereby all aspects of DevOps are measured using the CMMI model.

The DevOps Cube assessment is based on the idea that DevOps can be viewed from six different perspectives. Each gives a specific picture of the design of the DevOps philosophy. The fronts of the cube are based on 'The Three Ways': 'Flow', 'Feedback' and 'Continuous learning', the back of the cube include: 'Governance', 'Pipeline' and 'QA'.

The Continuous Everything assessment comprises six list of questions that make the DevOps maturity measurable on five levels. The following dimensions are included: 'Continuous Integration', 'Continuous Delivery', 'Continuous Testing', 'Continuous Monitoring', 'Continuous Documentation', and 'Continuous Learning'. This assessment book is an excellent mirror for every DevOps team that wants to quickly obtain a complete picture of the DevOps best practices to be addressed.

Author : Bart de Best
 Publisher : Leonon Media, 2019
 ISBN (NL) : 978 90 71501 814
 ISBN (UK) : 978 94 92618 047



DevOps Architecture

DevOps Architecture Best Practices

The world of system development is changing at a rapid pace. Development (Dev) and Operations (Ops) are being integrated more and more so that solutions can be offered to customers faster and with better quality. The question is how within this new view of DevOps there is room for Agile architecture. This book provides an answer to this question by giving many examples of architectural principles and models that give direction to the design and operation of a DevOps organization. Throughout the book an explanation is given as much as possible per paragraph based on an imaginary Assuritas company. This book consists of various parts, which makes the book modular. So, it does not have to be read from A to Z.

After the brief outline of the case company an explanation of how to construct the DevOps organization from an architectural perspective is given. The DevOps service management is then discussed. Both aspects are made clear on the basis of the case company. After explaining how the Dev and Ops roles can be integrated, two useful assessment tools to determine the maturity of DevOps are described. The book concludes with a case in which the choice for an 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 organization.

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



Agile Design

A set of best practices for an evolutionary design of information systems

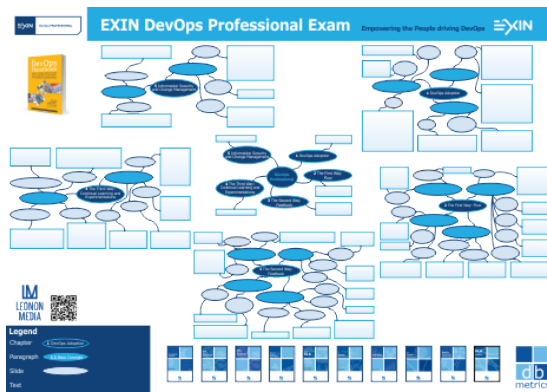
In recent years the need for a design for an information system has been questioned by many organizations. The classic justification for bundling information about an information system and thinking upfront about the contours of the information system to be realized are seen as obsolete by the Agile way of working and the idea of 'the three amigos development strategy'. Capturing the design of the information system (emerging design) during an Agile project is also omitted by many organizations. However, a design is also essential for safeguarding the transfer of knowledge, support for service management and compliance with laws and regulations. Elements that guarantee the continuity of an organization.

This book describes how an Agile design can be implemented so that both the elements for business continuity and flexibility are guaranteed. The design is divided into the following views: business, solution, design, requirements, test and code view.

The Agile 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 use of an Agile design lies in the realization of the information system, namely by integrating the Agile design, the test driven development and continuous documentation.

With this Agile approach to design, you have a powerful tool in your hands to get grip on an Agile development project.

Author : Bart de Best
 Publisher : Leonon Media, 2019
 ISBN (NL) : 978 94 92618 375
 ISBN (UK) : 978 94 92618 399



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.