Quality for DevOps teams

TMAP®

preview version

Rik Marselis
Berend van Veenendaal
Dennis Geurts
Wouter Ruigrok

sogeti
Part of Capgemini

Quality for DevOps teams

RIK MARSELIS
BEREND VAN VEENENDAAL
DENNIS GEURTS
WOUTER RUIGROK

sogeti
Part of Capgemini

Editing, typesetting & ebook production LINE UP boek en media bv, Groningen
Illustrations & cover design Axioma Communicatie, Baarn

ISBN 978 90 75414 89 9 (book)
978 90 75414 90 5 (ebook)

Foreword by Patrick Debois

Even though testers didn't make it to the word DevOps, they have always been an integral part of it. In the old days they would act as the wall between dev and ops, the final gate-keeper to production. Borrowing from systems theory, we want to optimize the whole system and not just the separate parts to achieve better results. This means quality goes beyond the testers group and is a responsibility for everyone. Everyone has to be taken in a very broad sense, not merely on the technical side; it also includes the process on how things are done and what is most important to the business. Sales, marketing and more recently DevRel (developer relations), can be seen as way to get valuable feedback too. If you optimize for quality, you will eliminate bottleneck after bottleneck. This is a noble goal, but one has to be focused on the business value. That balance is tough though: how much quality is enough? Speaking with the different groups inside the company plus, most importantly, the customer who buys and uses it, is crucial: it's all about gathering feedback to improve things.

In this book we see that modern organizations have changed: instead of relying on the strength of individuals, they aim for high-performing teams. By following "you build it, you run it", these teams put quality dead center because they would feel the pain of issues themselves. On their trip across the whole enterprise the authors show us how DevOps made an impact on the concept of quality in all different aspects of the daily work of an IT worker. It is easy to confuse DevOps with faster delivery where in fact it is as much about fast as it is about quality releases.

All the automation happening inside the delivery pipeline has not resulted in having less people involved or even having people fired. It has given humans time to work on things that matter. This continuous process is a learning experience and can only be achieved when we work together to make our jobs better.

Patrick Debois
Author of *The DevOps Handbook* and organizer of the first ever DevOps conference.

Foreword by the authors

Deliver value with the right quality at speed. That's what organizations ask of their IT teams. To achieve this, IT people need to work closely together. Cross-functional teams of business analysts, developers, testers, operations people, and other involved areas of expertise, join forces to work towards business objectives. Collaboration is key. Collaboration between people that have a shared responsibility and pick up the tasks that need to be done, based on competences and roles, no longer worrying about official functions. The DevOps culture is the enabler for this way of high-performance IT delivery, using an automated CI/CD pipeline to deliver at speed. But how do you deliver value with the right quality using DevOps? This book, and our website www.tmap.net, enable teams to implement quality engineering practices using our body of knowledge that contains various approaches, practical examples and concise explanations. We trust that teams will benefit from this and enhance their collaboration to deliver value to their stakeholders.
Aligned with the DevOps culture of IT delivery – where roles are important and functions are not – keep in mind that if in this book you see a term such as developer or tester, we refer to the role of a team member at a specific moment in time, not to a function.

Exploration is an important activity today because in high-performance IT delivery there's not enough time to first work out every detail; during the journey things need to be explored. As Apollo 8 astronaut Frank Borman put it: "Exploration is really the essence of the human spirit, and I hope we will never forget that." [Kluger 2017]
We encourage you to use this book in the same spirit. Explore the book, see what you can use. And adjust it to your own needs and those of the team.

When we created this book, we were inspired by many people. We also used a lot of knowledge from books, articles, conference-presentations, websites and other sources. Where we directly quoted from a source, we have put the reference to that source in the text. But, as you can imagine, we have acquired many ideas and knowledge over the years and can't exactly attribute these to a specific source. Therefore, we apologize to anyone

who is of the opinion that we reused their idea without a direct reference, we trust you will see this as a recognition and appreciation of your good work. We have done our best to make the list of references as complete as possible.

Although the name of Leo van der Aalst is not mentioned on the front cover of this book, Leo has written large parts of this book and was of great inspiration to the other authors. Before this book was finished, Leo had the opportunity to switch his career to a new direction beyond Sogeti, therefore he won't be able to support the other authors to convey the message in the future. The other authors are very grateful for his great contributions and collaboration.
We are grateful to all people that contributed during this endeavor (see our product owners' words for details). We thank our managers for enabling us to make this book a reality, thank you Rob Vijverberg, Christiaan Hoos and Tinus Vellekoop!

We trust you will find value in this book and wish you success and joy in applying the contents in your daily work.

Rik Marselis, principal quality consultant
Leo van der Aalst, DevOps lead
Berend van Veenendaal, expert backend developer
Dennis Geurts, lead software architect
Wouter Ruigrok, Agile quality coach

Acknowledgements by the product owner

Quality at speed is the challenge for IT delivery teams today. To support all people involved in facing this challenge, we embarked on a journey about a year ago to create an updated version of our body of knowledge, published on the website www.tmap.net, and a book that specifically deals with the challenges around quality and value a DevOps culture brings to organizations. Previously, TMAP was mainly for QA and testing professionals. In DevOps, the delivery of a product, including the quality of that product, is the responsibility of the team as a whole, therefore our body of knowledge supports all people involved: the DevOps team members and also the other people in the organization, with knowledge, skills, tools and more.
In our project, which started in the Netherlands and quickly involved people across many countries, we adhered to the principles of DevOps; we shared the responsibility of creating this book, for example.
Open source is widely adopted today, we have always supported this idea of sharing knowledge so that all can benefit. The TMAP body-of-knowledge is open and available to anyone to use. And I also encourage you to contribute your examples, good practices, building blocks or whatever can be useful, so that we will build a strong quality engineering community together.

Our journey really started with a kickoff meeting in April 2019 where an international group of IT experts came together and spent two days to lay the foundations of this new view on high-performance IT delivery.
The participants were Robbert van Alen, Eveline Moolenaars-Koetsier, Bert Linker, Wouter de Kort, Gitte Ottosen (DK), Nicklas West (SE), Vincent Wijnen, Filip Joele, Joey van Hoek, Jurian de Cocq van Delwijnen and Jan Sleutjes. As a result of this meeting (which was organized and moderated by Rik Marselis and Leo van der Aalst), the team decided on the relevant QA and testing topics, created various definitions and defined the important starting points to further work out our view on high-performance IT delivery.

We had a very inspiring meeting with the following representatives of our clients: Paul Custers (NS), Werner Soeteman (KLM), Willem-Jan van Tongeren (PostNL), Reindrich Geerman (KPN), Edward Elgeti (UWV), Peter Claassen (Rabobank) and Caroline Arkesteijn (NIBC Bank). We are grateful for them clarifying their business challenges. They confirmed the direction to which we were headed, pinpointed specific subjects of interest, shared practical insights and made a number of valuable nuances.

Many people contributed to the contents of this book and the body of knowledge by inspiring the authors and by presenting useful examples, snippets of text and references to valuable sources.
For this we thank (in no particular order) Eva Holmquist (SE), Peter Betting (FR), Tom van de Ven, Andrew Fullen (UK), Barry Weston (UK), Fredrik Scheja (SE), Marianne Duijst, Paul van de Geer, Matthias Hamburg (DE), Sven Fanslau (DE), René Boswinkel, Arno Balemans, Albert Tort Pigubet (ES), Geert Vanhove (BE), Geert Jan Carpay, Gijs Op de Beek, Maurice Siteur, Marc Roekens, Martijn van der Salm, Erwin Riemersma, Mark van der Walle, Emil Wesselink, Charlotte Janus, Rianne Oorebeek – de Neef, Jan Sleutjes, Dirkjan Kaper, Jan-Willem van den Brink, Tinus Vellekoop, Marco van den Brink and Casper Schipper.

I would like to extend special recognition to Richard Ammerlaan, Randy Semeleer, Bas de Heer and Daan Kroese who largely contributed to the chapters about non-functional quality.
We also received contributions from the quality & testing community; thank you Derk-Jan de Grood, Ard Kramer and Bart Broekman. I would also like to recognize all participants to the 24th Testing Retreat for sharing their visions on this topic. The Testing Retreat is an annual independent informal meeting of international QA & testing experts.

When the book neared its completion, a great number of people invested their valuable time in reviewing the semi-finished text blocks and the complete manuscript. We especially thank the clients and partners that helped us: René Tuinhout (RDW), John Bertens (Achmea), Bart Enkelaar (Bol.com), Pascal Nicolakis (Micro Focus), Arno van der Velde (Micro Focus) and Patrice Chorot (Micro Focus).
Our colleagues contributed a great number of suggestions, comments and improvements, thank you Albert Tort Pigubet (ES), Andrew Fullen (UK), Antoine Aymer (FR), Barry Weston (UK), Blue Ityalam (IR), Eva Holmquist (SE), Fethi Mebrouk (FR), Gitte Ottosen (DK), Jürgen Beniermann (DE), Maheshwar Kanitkar (IN), Markus Niehammer (DE), Matthias Hamburg (DE), Mona Iversen (NO), Monish Pawar (IR), Nicklas West (SE), Peter Betting (FR), Sven Fanslau (DE), Torunn-Cathrine Ludvigsen (NO), Vishal Rai (IR), Arno Balemans, Bert Linker,

Clemens Reijnen, Dirkjan Kaper, Erik Kuipers, Eveline Moolenaars-Koetsier, Filip Joele, Geert Jan Carpay, Gert Stad, Gijs Op de Beek, Hester van der Helm, Marco Jansen van Doorn, Marianne Duijst, Mark van der Walle, Maurice Siteur, Paul van de Geer, Pepijn Paap, Ralph Klomp, René Boswinkel, Richard Ammerlaan, Robbert van Alen, Stefan Gerstner and Mark Buenen.
Organizing this review process and merging all feedback was a job very conscientiously done by Sander van Logchem.

In the process of creating a book there are some activities that need to be done by specialists. We are very happy with the support we got from our publisher, LINE UP boek en media, in particular Peter ten Hoor, Minke Sikkema, Mirjam Kroondijk and Annelies Gallagher. The professional look and feel of this book can all be attributed to the Axioma design team, Richard Wouters, Lex Stuip and Pepijn van der Meer, and to our colleagues of marketing & communications Linda van Tilborgh, Jolien Dusseldorp-Schipper and Nicolette van der Heide.

Last but most certainly not least I would like to thank the authors of this book. Creating a book like this is a long journey. A journey that isn't always a smooth ride; the road can be bumpy or even rocky sometimes, a journey of taking one step forward and two steps back by occasion. They persisted in achieving their goal to create a book accessible to all members in a high-performance team. Hats off for Rik Marselis, Berend van Veenendaal, Dennis Geurts and Wouter Ruigrok! Of course, special recognition for Leo van der Aalst and Richard Ammerlaan who contributed by supplying substantial content for this book.

I trust you will enjoy reading this book and hope you will be inspired to apply the knowledge and use the insights and tools to the benefit of your IT delivery process.

Rob Vijverberg, Head of Digital Assurance and Testing, Sogeti NL
Vianen, February 2020

Table of contents

Foreword by Patrick Debois 3
Foreword by the authors 4
Acknowledgements by the product owner 6

Part 1 Introduction 15

1 What value will you find in this book? 16
- 1.1 The DevOps IT delivery model 17
- 1.2 Continuous quality engineering 18
- 1.3 README.TXT (reading guide) 21
- 1.4 TMAP evolution 23

2 Successful high-performance IT delivery depends on people 26

3 The VOICE model 34

4 Examples of indicators in the VOICE model 39

5 Introduction to quality and testing 42

6 CI/CD pipelines and tooling 53

Part 2 IT delivery models 61

7 Overview of IT delivery models 62

8 Sequential IT delivery models 66
- 8.1 Waterfall 66
- 8.2 V model 68

9 High-performance IT delivery models 70
- 9.1 Scrum 71
- 9.2 DevOps 73
- 9.3 Other high-performance IT delivery “models” 78

10 Hybrid IT delivery models 85
- 10.1 Demand/supply model 86
- 10.2 SAFe model 89

Part 3 QA & testing topics 93

11 Introduction to QA & testing topics 94
12 Introduction to organizing QA & testing topics 98
13 Introduction to performing QA & testing topics 101
14 Topics plotted on the IT delivery models 103
14.1 Sequential IT delivery models 103
14.2 High-performance IT delivery models (especially DevOps) 104
14.3 Hybrid IT delivery models (especially SAFe) 107

Part 4 Organizing topics explained for DevOps 111

15 Quality & test policy 112
16 Responsibilities & roles 118
17 Monitoring & control 124
18 Anomaly management in DevOps 130
19 Reporting & alerting 135
20 Estimating the effort 145
21 Planning the delivery 155
22 Infrastructure 165
23 Tooling 171
24 Metrics 181
25 Continuous improvement 193

Part 5 Performing topics explained for DevOps 205

26 Quality risk analysis & test strategy 206
27 Acceptance criteria 216
28 Quality measures 219
29 Reviewing (static testing) 222
30 Test design 229
31 Test data management (TDM) 231
32 Test automation 238
33 Test execution 249
34 Investigate and assess the outcome of testing 252

Part 6 Quality measures and skills 257

35 Description of quality measures 258

35.1 Root cause analysis 258
35.2 Specification and Example (SaE) 259
35.3 Test-driven development 265
35.4 Pair programming 267
35.5 Pairing 268
35.6 Reviewing 269
35.7 Test design techniques 270
35.8 Feature toggles 271
35.9 Monitoring of product quality 273
35.10 Parallel testing 274

36 Personal, interpersonal and team skills 276

Part 7 Test varieties 285

37 Make sure there is variety in your testing 286
38 Performance testing 294
39 Usability testing 306
40 Security testing 314
41 Maintainability testing 321
42 Mutation testing tests the test 325

Part 8 Test design 331

43 The many aspects of creating tests and exploring a test object 332

43.1 Two distinct ways of preparing and performing tests 333
43.2 Experience-based testing 333
43.3 Coverage-based testing 334
43.4 Always combine experience- and coverage-based testing 337

44 Test design entities 338
45 Test approaches applied 343

45.1 Coverage-based testing 343
45.2 Process-oriented test design 345
45.3 Condition-oriented test design 345
45.4 Data-oriented test design 346
45.5 Appearance-oriented test design 347
45.6 Selecting and combining approaches and techniques 347

46 There are many techniques, which one to use? 350
46.1 No risk – no test – no development 352
46.2 Example case: “The magic boat ride in TestLand” 352
46.3 Process-oriented testing: path coverage 353
46.4 Condition-oriented test design 358
46.5 Data-oriented testing with EP and BVA 362
46.6 Data combination test 366
46.7 Syntactic testing 370
46.8 Code coverage in many different intensities 371
47 Experience-based testing 376
47.1 Experience-based testing approaches 376
47.2 Checklist 379
47.3 Error guessing 381
47.4 Exploratory testing 383
47.5 Crowd testing 390
47.6 What are heuristics and when to use them? 394
48 Is there any value in unstructured testing? 398

Appendix: Quality characteristics and non-functional testing 404
References 415
Index 423

Part 1
Introduction

1

What value will you find in this book?

Organizations can only be successful when properly supported by IT systems. People involved in the creation and use of IT systems face many challenges. Updates and changes have to be delivered faster than ever before. The right quality is vital to maintain service in our 24/7 economy. User experience should be right first time to retain customers. Teams need to deal with these challenges, integrating quality measures throughout the IT delivery process and using tools to deliver at speed.

Information Technology (IT) has made giant technological leaps forward and still evolves at an incredible pace. The way IT systems are created and delivered has also evolved. Today, IT systems are preferably delivered incrementally with frequent adaptation to new business needs and user expectations, and with increasing speed, supported by state-of-the-art tools. We should keep a close eye on what it is all about: delivering software – which will generate business value – at the right time with the desired quality!

This book provides information on how to build in quality and establish a level of confidence so that the pursued business value can be achieved, as described in our VOICE model.

We call the modern iterative-incremental way of creating IT systems: high-performance IT delivery.

> High-performance IT delivery is an approach that enables cross-functional teams to continuously improve the products, processes and people that are required to deliver value to the end users.

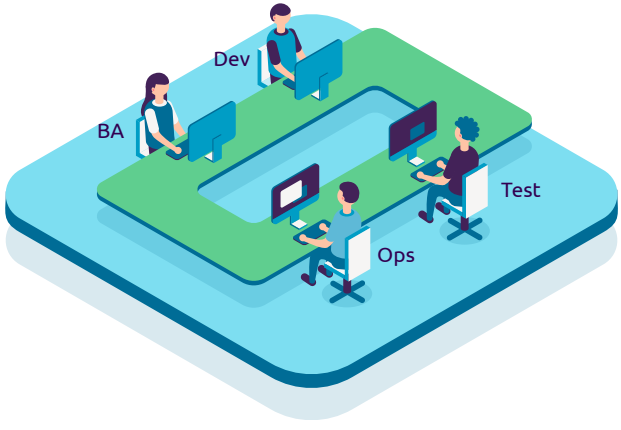
The title – *Quality for DevOps teams* – demonstrates that this book is primarily meant for those that want to deliver quality solutions in a DevOps setting. Although IT is often about tools and techniques, people still make the difference between dissatisfaction or success. Do you work in a high-performance IT delivery approach, such as DevOps, and do you want to make sure you deliver value to users by assuring the right level of quality? And do you want to establish the level of confidence that the pursued business value can be achieved? Then you will find great benefit in the valuable ideas, approaches and experiences in this practical book.

1.1 The DevOps IT delivery model

Since DevOps is the main IT delivery model in this book, let us briefly describe DevOps.

DevOps is a cross-functional systems engineering culture that aims at unifying systems development (Dev) and systems operations (Ops) with the ability to create and deliver fast, cheap, flexible and with adequate quality, whereby the team as a whole is responsible for the quality. Other areas of expertise, such as business analysis and quality assurance (including testing) are usually integrated in the team. A DevOps culture has an Agile mindset that can be supported/implemented by, for example, the Scrum framework.

Figure 1.1 DevOps cross-functional team.

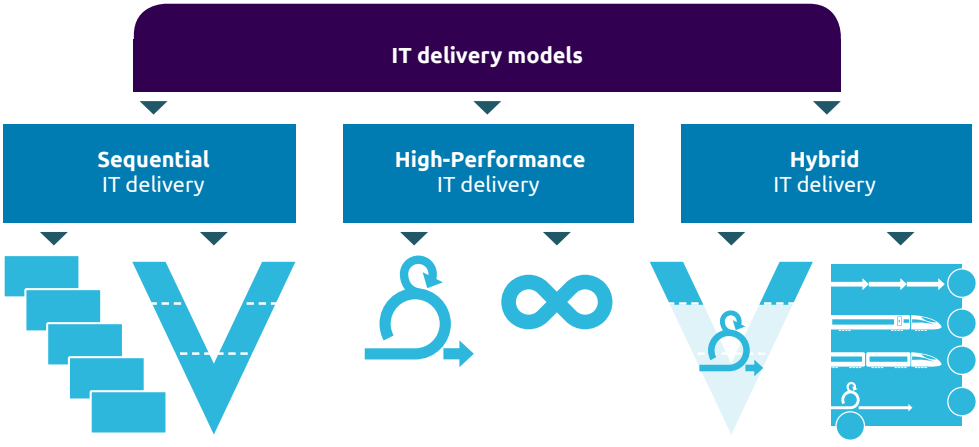


Working in a cross-functional team means that the team as a whole is responsible for delivering value. The team has all competencies and skills to perform the necessary tasks and no team member has the monopoly on performing any task. This way the team can always go forward, even when a team member is temporarily not available. And of course, a team can work together with specialists from other teams or support groups for specific tasks.

Overview of IT delivery models

In addition to DevOps, Scrum is also considered one of the high-performance IT delivery models. Besides the high-performance IT delivery models, we distinguish two other IT delivery models (Figure 1.2): Sequential models (e.g. waterfall and V model) and a blend of sequential and high-performance IT delivery models, resulting in a hybrid IT delivery model (e.g. demand/supply and SAFe). You can read more about IT delivery models in Chapter 7, "Overview of IT delivery models".

Figure 1.2 IT delivery models.



1.2 Continuous quality engineering

In the DevOps IT delivery model, there is continuous focus on quality engineering. Actually, commonly DevOps teams try to implement "continuous everything", which means that they strive to automate as many tasks and activities as possible. This leads to, among

other things, Continuous Integration and Continuous Deployment (commonly abbreviated to CI/CD).
To implement continuous quality engineering, of which continuous testing is a part, DevOps teams must use state-of-the art tools powered by artificial intelligence and machine learning. This will enable them to deliver quality at speed, for example by forecasting quality problems and solving them before anyone experiences a failure.

1.2.1 Vision: "built-in quality"

We strongly believe that high-performance IT delivery teams have capabilities as a team, the capabilities of the people in the team contribute to the team as a whole, and tasks can be performed by various team members. One team member may be more skilled than another, but that is no reason to only provide work to the team member with the most experience and skills. Think of all tasks as a team effort in which each team member plays a role. Specifically, we think that quality assurance (QA) and testing activities should be integrated in both the DevOps activities as well as the people involved.

1.2.2 DevOps activities

DevOps is an elusive phenomenon. There are discussions about who is the founder of DevOps and what the DevOps IT delivery model looks like exactly. To avoid this discussion, we use a simplified model in this book, on which the common activities are plotted (Figure 1.3).

Figure 1.3 The six DevOps activities.

We identify six DevOps activities:

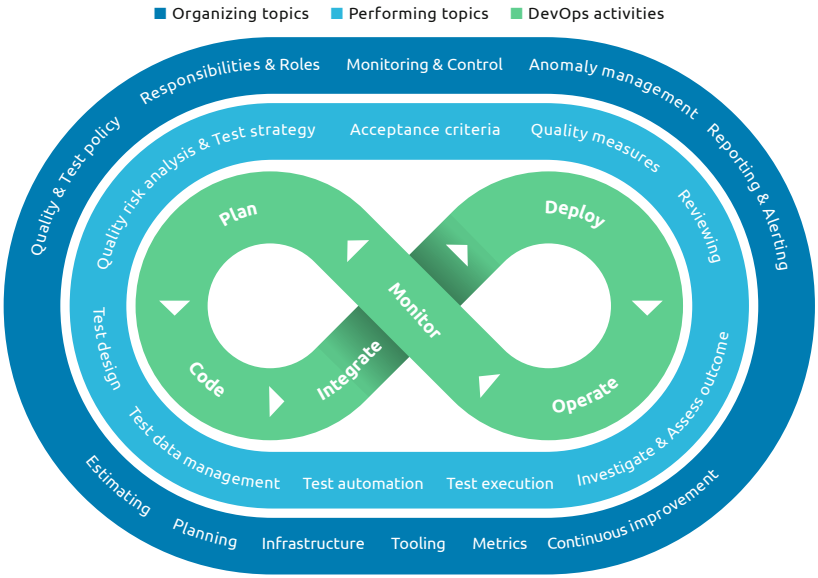
- Monitor
- Plan
- Code
- Integrate
- Deploy
- Operate

These activities provide excellent support to explain the integration of DevOps activities with the QA and testing topics. For more information on the DevOps activities see the description in Chapter 9, “High-performance IT delivery models”.

1.2.3 Quality assurance & testing topics

In this book, we take a list of quality assurance & testing activities, grouped in “Organizing” and “Performing” topics (Figure 1.4), as our starting point. We describe how these topics relate to the DevOps activities and what the people involved need to know and need to do related to these various topics.

Figure 1.4 Quality assurance & testing topics and DevOps activities.



The topics are not directly and one-on-one related to the DevOps activities. Depending on the activity, you will see that one or more topics can relate to a specific activity. See Chapter 14, "Topics plotted on the IT delivery models".
These topics can – or even should – be applied by all DevOps people. Not just by a tester or a person with a quality assurance role. In general, quality assurance and testing topics are integrated with all DevOps activities and executed by all people involved.
This book will support you in organizing and performing quality engineering tasks within DevOps, irrespective of whether you see yourself as a developer, an operations person, a business analyst, a quality engineer, a tester, a product owner, a coach, a manager or any other role or function.

1.3 README.TXT (reading guide)

Quality for DevOps teams is not a book to read from cover to cover. Also it is not intended as a step by step implementation of quality engineering activities in DevOps. Instead, this book provides relevant parts of the TMAP body of knowledge. Depending on your needs, interests and experience with quality engineering, parts can be read carefully, scanned quickly, or even skipped altogether.
The book is intended for all people involved in DevOps. But every person will want to use another subset from this book. The parts, chapters and sections provide a handy division. The table of contents at the beginning of the book and the index at the end of the book will also assist you in locating relevant knowledge.
You will find an extensive glossary with over 300 terms on the website. The relevant definitions are included in this book and are clearly shown in a textbox. If you are looking for a specific definition, please use the index to find the page where the term is explained or find the glossary on www.tmap.net.

1.3.1 Overview of the parts of the book

This book is divided into eight parts. A short overview:

- *Part 1 Introduction*
  In this part, we explain the reason why you should read this book, that successful high-performance IT delivery depends on people; we explain the VOICE model (confidence in pursued value), and give an introduction to quality assurance & testing. At the end of this part we describe the CI/CD pipeline.

- *Part 2 IT delivery models*
  In this part, we provide an overview of three IT delivery models: sequential, high-performance and hybrid. And, since this is the focus of our book, we provide an in-depth explanation of the high-performance IT delivery model DevOps.
- *Part 3 QA & testing topics*
  In this part, you will find an introduction to QA & testing topics, which are grouped into "Organizing" and "Performing" topics. Examples of cross references between these topics and the three IT delivery models can be found in this part as well.
- *Part 4 Organizing topics explained for DevOps*
  In this part, the eleven organizing topics are described (Figure 1.4). These organizing topics are aimed at arranging, planning, preparing and controlling QA & testing activities.
- *Part 5 Performing topics explained for DevOps*
  In this part, the nine performing topics are described (Figure 1.4). These performing topics are aimed at the operational, specifying and executing QA & testing activities.
- *Part 6 Quality measures and skills*
  In this part, a wide variety of quality measures is described. These can be used independently as part of quality assurance as well as to cover risks. In this part, you will also find a description of personal, interpersonal and team skills.
- *Part 7 Test varieties*
  In this part, we explain the concept of test varieties and describe four test varieties in more detail. The testing pyramid and the testing quadrants are also discussed.
- *Part 8 Test design*
  In this part, you will find a description of experienced-based testing and its four approaches (checklist, error guessing, exploratory testing, crowd testing). We also provide an explanation of coverage-based testing, its four groups (process, condition, data, appearance) and at least one application of a test design technique for each group.

1.3.2 Target groups of this book

The main target group of this book is people working in DevOps teams. Others with an interest in quality engineering in today's IT delivery will also find a lot of interesting knowledge in this book.

If you have never read anything about TMAP, this book certainly will help you get an overview of what it has to offer. On the other hand, if you are experienced in applying TMAP in your work, you will notice that we have made significant additions and adaptations to make it completely fit in today's IT world.

As an aid to choosing the parts that are of interest in connection with certain roles, we also provide you with reading suggestions.
In general, our suggestion is first to determine the challenges in your own situation, and then select the parts, chapters and topics in this book as a source of inspiration and information to implement improvements.
Since every person has a different need for information, we cannot specify which parts of this book are of interest to you. However, the following hints and tips may help you find your way in this book.
We advise everyone to start with Part 1 to understand the basic starting points.
If you are interested in different ways of IT delivery read Part 2, if you are only interested in DevOps, you may skip this part.
QA & testing consists of many activities; we have structured these activities in twenty topics as described in Part 3.
If you are involved in organizing QA and testing in DevOps (for example as a product owner, agile coach, test manager etc.), we refer to Part 4.
If you are involved in performing QA and testing in DevOps (for example as a business analyst, a developer, a tester or an operations person), we refer to Part 5.
Part 6 is valuable as a reference to specific quality measures; please select from the table of contents the quality measures that are relevant in your situation.
If you set up quality engineering, you have to make sure there is sufficient variety in testing; Part 7 will give you the information needed.
If you are involved in specifying and executing tests, you will find very useful information about test design and execution (and the test design techniques that TMAP is famous for) in Part 8.
If you need to define the focus of quality engineering, you will find an overview of quality characteristics in the appendix.

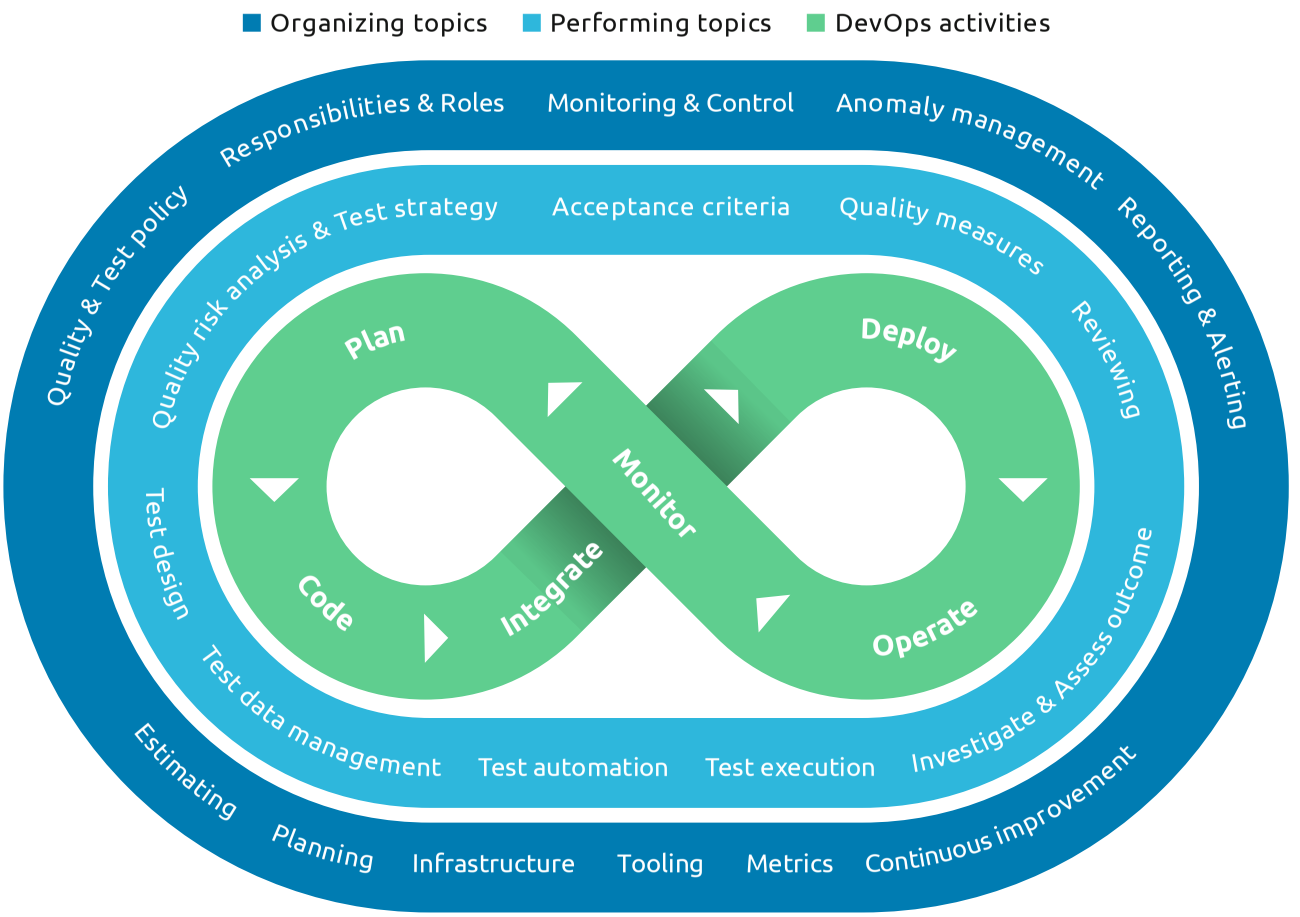
1.4 TMAP evolution

TMAP is a body of knowledge for quality engineering in IT delivery. The strength of TMAP can be largely attributed to the considerable practical experience that is the basis for the body of knowledge. This experience comes from thousands of IT professionals in as many projects over the last twenty-five years.
TMAP was created in 1995 and back then mainly contained a process description for testing. Over the years, TMAP has evolved from process-driven, business-driven and human-driven to a body of knowledge with which quality assurance & testing activities can be

integrated in all IT delivery models. The body of knowledge contains a wide range of approaches, good practices, techniques and tools for organizing and performing of activities related to quality assurance and testing of IT systems.
Today we use TMAP as a term on its own, rather than a meaningful abbreviation. For this reason TMAP is now written with four capital letters.

TMAP website

The TMAP body of knowledge is easily accessible on the website www.tmap.net. There are also numerous books available in the TMAP series. Books are static, IT delivery landscapes are changing faster and faster. We therefore aim to keep the website up to date. You can find most of the content of this book on this website, as well as many in-depth explanations, additions to the content of this book and up-to-date overviews of relevant tools.
In addition, the quality assurance & testing topics are worked out for other IT delivery models.
In summary, this book is supportive of the website, which is constantly updated and adapted to changed IT delivery landscapes and new insights.
We wish you a lot of useful insights and good luck with applying TMAP knowledge in your situation.



Quality assurance & testing topics and DevOps activities

This is a preview version.

Please refer to the full version for chapters 2 through 47.

References

[Aalst 2008] TMap NEXT Business Driven Test Management, Leo van der Aalst, Rob Baarda, Ewald Roodenrijs, Johan Vink, Ben Visser, 2008, UTN Publishers, ISBN 90-72194-93-4

[Aalst 2010] Testing expensive? Not testing is more expensive!, Leo van der Aalst, Corné de Koning, 2010, http://leovanderaalst.nl/Article%20Testing%20expensive%20-%20Not%20testing%20is%20more%20expensive.pdf

[Aalst 2012] TMap NEXT in Scrum – Effective testing in Agile projects, Leo van der Aalst, Cecile Davis, 2012, Sogeti Nederland, ISBN 978-90-75414-64-6

[Aalst 2016] Beyond I-, T-, Π- and m-shaped-testers, Leo van der Aalst, 24 October 2016, https://labs.sogeti.com/beyond-t-Π-m-shaped-testers/

[Adzic 2011] Specification by Example: How successful teams deliver the right software. Adzic, Gojko, 2011, Manning. ISBN 9781617290084.

[Agile Manifesto 2001] Manifesto for Agile Software Development, Kent Beck et al, 2001, https://agilemanifesto.org/

[Altexsoft 2019] Acceptance Criteria: Purposes, Formats, and Best Practices, Altexsoft, accessed November 2019, https://www.altexsoft.com/blog/business/acceptance-criteria-purposes-formats-and-best-practices/

[Balaji 2012] Waterfall versus V-Model, A Comparative Study on SDLC, S. Balaji, 2012, http://www.jitbm.com/Volume2No1/waterfall.pdf

[Bangser 2019] Observability, Abigail Bangser, May 2019, https://club.ministryoftesting.com/t/power-hour-curious-stuck-or-need-guidance-on-devops-or-observability/24963/17/

[Basili 1994] The Goal-Question-Metric Approach, Basili, V.R., Galdiera, G. and Rombach, H.D., 1994, John Wiley & Sons, Inc., New York

[Beck 2002] Test-Driven Development By Example, Kent Beck, 2002

[Beizer 1995] Black-Box Testing – Techniques for Functional testing of Software and Systems, Boris Beizer, 1995, John Wiley & Sons inc., ISBN 978-0-471-12094-0

[Black 2004] What IT Managers Should Know about Testing: How to Analyze the Return on the Testing Investment, Rex Black, 2004, www.rbcs-us.com/site/assets/files/1345/what-it-managers-should-know-about-testing-roi.pdf

[Black 2017] The expert test manager, Rex Black, Leo van der Aalst, James L. Rommens, 2017, Rockynook, ISBN 978-1-93395-294-9

[Boersma 2014] Neil's Quest for Quality – A TMap HD story, Aldert Boersma, Erik Vooijs, 2014, Sogeti Nederland, ISBN 978-90-75414-83-7

[Bolton 2019] tweet about exploration, Michael Bolton, twitter 17 July 2019, 23:35

[Broekman 2003] Testing embedded software, Bart Broekman, Edwin Nooteboom, 2003, Addison Wesley, ISBN 978-0-321-15986-1

[Buwalda 2004] Soap opera testing, Hans Buwalda, February 2004, Better software magazine

[Clokie 2017] A practical guide to testing in DevOps, Katrina Clokie, 2017, Leanpub

[Cockburn 2000] The Costs and Benefits of Pair Programming, Cockburn, Alistair and Williams, Laurie, Humans and Technology Technical Report 2000.01, Jan 2000, http://alistair.cockburn.us

[Cohn 2009] The Forgotten Layer of the Test Automation Pyramid, Mike Cohn, 17 December 2009, https://www.mountaingoatsoftware.com/blog/the-forgotten-layer-of-the-test-automation-pyramid

[Crispin 2008] Agile Testing – A Practical Guide for Testers and Agile Teams, Lisa Crispin & Janet Gregory, 2008, Addison Wesley, ISBN 978-0-321-53446-0

[DASA 2019] DASA DevOps Principles, DevOps Agile Skills Association, website accessed November 2019, www.devopsagileskills.org

[Dinwiddie 2009] If you don't automate acceptance tests?, George Dinwiddie, 2009, http://blog.gdinwiddie.com/2009/06/17/if-you-dont-automate-acceptance-tests/

[DORA 2019] DevOps Research and Assessment, Nicole Forsgren, Jez Humble, Gene Kim and Soo Choi, 2019

[Doran 1981] There's a S.M.A.R.T. way to write management goals and objectives, George Doran, Arthur Miller and James Cunningham, 1981, article

[EuroSTAR 2017] The little book of Testing Wisdom, various authors, 2017, EuroSTAR software testing

[Forrester 2019] Forrester Research on DevOps: Quality Metrics that Matter, Tricentis, https://www.tricentis.com/resources/forrester-research-on-devops-quality-metrics/

[Fowler 2004] Specification By Example, Martin Fowler, 18 March 2004, https://martinfowler.com/bliki/SpecificationByExample.html

[Fowler 2017] Feature Toggles (aka Feature Flags), Martin Fowler, 9 October 2017, https://martinfowler.com/articles/feature-toggles.html

[Gherkin 2020] Keywords, Cucumber website, accessed 2020, https://cucumber.io/docs/gherkin/reference/#keywords

[Guest 1991] The hunt is on for the Renaissance Man of computing, Davis Guest, The Independent, 17 September 1991

[Hamburg 2019] A domain model for testing, Matthias Hamburg, German Testing Magazine 01-2019, January 2019

[Hendrickson 2013] Explore It!, Elisabeth Hendrickson, 2013, the Pragmatic Programmers, ISBN 978-1-93778-502-4

[Humble 2016] The DevOps Handbook, John Willis, Patrick Debois, Jez Humble, Gene Kim, October 2016, IT Revolution Press, ISBN 978-1457191381

[IDEA 2009] White paper Usability Testing according to TMap NEXT, Sogeti Workgroup IDEA: Usability testing, december 2009, Sogeti Nederland

[IEEE1044 2009] Standard Classification for Software Anomalies, Institute of Electrical and Electronics Engineers, 2009, https://standards.ieee.org/standard/1044-2009.html

[IEEE1028 2008] IEEE Standard for Software Reviews and Audits, Institute of Electrical and Electronics Engineers, 2008, https://standards.ieee.org/standard/1028-2008.html

[ISO25010 2011] Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models, International organization for standardization, 2011, ISO, www.iso.org

[ISTQB 2019] Standard Glossary of Terms used in Software Testing version 3.2, ISTQB, www.istqb.org

[ITIL 2019] What is ITIL, Axelos, accessed November 2019, https://www.axelos.com/best-practice-solutions/itil/what-is-itil

[Jeffries 2001] Essential XP: Card, Conversation, Confirmation, Ron Jeffries, 30 August 2001, https://ronjeffries.com/xprog/articles/expcardconversationconfirmation/

[Juran 1951] Juran's quality handbook, Joseph M. Juran, 1951, McGraw Hill

[Kahn 1990] Psychological Conditions of Personal Engagement and Disengagement at Work, Kahn, William A., 01-12-1990-12-01, Academy of Management Journal, 33 (4): 692–724. doi:10.2307/256287. ISSN 0001-4273. JSTOR 256287.

[Kano 2011] Attractive quality and must-be quality, Kano Noriaki, Nobuhiku Seraku, Fumio Takahashi, Shinichi Tsuji, 1984, Journal of the Japanese Society for Quality Control (in Japanese). 14 (2): 39–48. ISSN 0386-8230

[Kim 2012] The Three Ways: The Principles Underpinning DevOps, Gene Kim, 22 August 2012, https://itrevolution.com/the-three-ways-principles-underpinning-devops/

[Kim 2014] The Phoenix project, Gene Kim, Kevin Behr, George Spafford, 2014, IT Revolution press, ISBN 978-0-9882625-0-8

[Kluger 2017] Apollo 8, Jeffrey Kluger, 2017, Picador paperback, ISBN 978-1-250-18251-7

[Koomen 2006] TMap Next for result-driven testing, Tim Koomen, Leo van der Aalst, Bart Broekman, Michiel Vroon, 2006, UTN publishers, ISBN 90-72194-80-2

[Lerner 2017] AIOps platforms, Andrew Lerner, 9 August 2017, https://blogs.gartner.com/andrew-lerner/2017/08/09/aiops-platforms/

[Marick 2003] My Agile testing project, Brian Marick, 21 August 2003, http://www.exampler.com/old-blog/2003/08/21.1.html#agile-testing-project-1

[Marselis 2007] TestGrip – gaining control on IT quality and processes through test policy and test organisation, Rik Marselis, Jos van Rooyen, Chris Schotanus, Iris Pinkster, 2007, LogicaCMG, ISBN 987-90-71195-01-3

[Marselis 2012] the PointZERO vision – stop wasting time and money, Rik Marselis, Ewald Roodenrijs, 2012, Sogeti Nederland, ISBN 978-90-75414-55-4

[McCracken 1957] Program Checkout in Digital Computer Programming, D.D. McCracken, 1957, https://www.quora.com/Why-does-Kent-Beck-refer-to-the-rediscovery-of-test-driven-development-Whats-the-history-of-test-driven-development-before-Kent-Becks-rediscovery

[Myers 1979] The art of software testing, Glenford J. Myers, 1979, Wiley Interscience, ISBN 0-471-04328-1

[Nielsen 1994] 10 Usability Heuristics for User Interface Design, Jakob Nielsen, 1994, https://www.nngroup.com/articles/ten-usability-heuristics/

[OWASP 2019] About The Open Web Application Security Project, OWASP, 21 August 2019, https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project

[PLT 2004] Principles of Lean thinking, Tools & Techniques for advanced manufacturing, National Research Council Canada, 2004, http://www.itc.mb.ca/downloads/resources_by_topic/princ_lean%20thinking/PrinciplesofLeanThinkingRevD2004.pdf

[Pyhäjärvi 2019] Mob programming guidebook, Maaret Pyhäjärvi, 2018, https://mobprogrammingguidebook.xyz/Download/MobProgramming-Guidebook.pdf

[Ravlani 2017] resources_by_topic/princ_lean%20thinking/PrinciplesofLean-ThinkingRevD2004.pdf

[Ryber 2007] Essential Software Test Design, Torbjörn Ryber, 2007, Fearless consulting, ISBN 978-91-85699-03-2

[SAFe 2019] Scaled Agile Framework version 4.6, 2019, https://www.scaledagileframework.com/

[Scrum 2017] The Scrum Guide; The Definitive Guide to Scrum: The Rules of the Game, Ken Schwaber and Jeff Sutherland, November 2017, https://scrumguides.org/docs/scrumguide/

[SIG/TÜViT 2018] Evaluation Criteria Trusted Product Maintainability version 10.0, prof.dr.ir. Joost Visser, Dennis Bijlsma MSc, dr. Haiyun Xu, 9 May 2019, Software Improvement Group

[Siteur 2019] Implementing Test Automation, Maurice Siteur, February 2019, Leanpub, https://leanpub.com/ImplementationTestAutomation

[Smit 2009] End-to-end testing with TMap NEXT, Rob Smit & Rob Baarda, 2009, UTN publishers, ISBN 90-72194-96-9 / 978-0-72194-96-1

[Smith 2020] Chasing Mutants (chapter in The Future of Software Quality Assurance), Adam Leon Smith, 2020, Springer, ISBN 978-3-030-29509-7

[Sogeti 2009] TPI NEXT Business Driven Test Process Improvement, Gerrit de Vries, Ben Visser, Loek Wilhelmus, Alexander van Ewijk, Marcel van Oosterwijk, Bert Linker, UTN Publishers, ISBN 978-90-72194-97-8

[Sogeti 2020] The six quality key areas are borrowed from Sogeti's Agile Quality Improvement framework, will be published in 2020

[STX Next 2019] What is NoOps?, STX Next, 25 April 2019, https://stxnext.com/blog/2019/04/25/aiops-bizdevops-cloudops-devops-itops-noops-introduction-digital-business-transformation/#what-is-noops

[Techopedia 2020] www.techopedia.com/definition/2/cloud-computing (accessed in February 2020)

[TMAP 2020] www.tmap.net

[Tort 2019] (Agile) development needs (agile) testing and on-team-vision, Albert Tort, 20 August 2019, SogetiLabs blog, https://labs.sogeti.com/wp-content/uploads/2019/08/10-tips-for-implementing-agile-development-with-quality.pdf

[Tuinhout 2008] The Boundary Value Fallacy, René Tuinhout, 2008, Testing Experience magazine

[Veld 1988] Analysis of organizational problems, an example of system and process oriented thinking, In 't Veld, 1988

[Ven 2018] Testing in the digital age – AI makes the difference, Tom van de Ven, Rik Marselis, Humayun Shaukat, 2018, Sogeti Nederland, ISBN 978-90-75414-87-5

[Wake 2003] INVEST in Good Stories and SMART Tasks, Bill Wake, 17 august 2003, https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/

[Webopedia 2020] www.webopedia.com/TERM/C/containerization.html (accessed in February 2020)

[Weinberg 2008] Perfect Software and other illusions about testing, Gerald M. Weinberg, 2008, Dorset House, ISBN 978-0-932633-69-9

[Wittig 2016] Amazon Web Services in Action, Andreas Wittig and Michael Wittig, 2016, Manning Press, ISBN 978-1-61729-288-0

[WQR 2019] World Quality Report 2019, Sogeti, Capgemini, Micro-Focus, 2019, https://www.sogeti.com/explore/reports/world-quality-report-2019/

[WSJF 2009] The Principles of Product Development Flow: Second Generation Lean Product Development, Donald G. Reinertsen, 2009, Celeritas, ISBN 1935401009, 9781935401001

Index

The **bold** page numbers refer to the page with the definition of that term.

100% code coverage 372

A

acceptance criteria **52**, 102, **217**
acceptance test 293
acceptance test-driven development 213, 259, 267
actual outcomes 252
Agile acceptance testing 259
Agile at scale 85
Agile Manifesto 26, 80
Agile mindset 26, 65, 78, **200**
Agile planning approaches 155
Agile principles 81
AIOps 32
alerting 136
alerts 144
algorithm test 355
amigos approach 224
anomaly 45, **130**, 141, 254
anomaly administration 132
anomaly management 99, 130, 134
anomaly report 255
APIs 288
appearance-oriented test design 347
artificial intelligence 19, 234
ATDD 259
auditing 127
auto-generation of test cases 242
automate everything 32
automated tests 56
automation framework 177
automation rate 40

B

BDD 259
behavior-driven development 213, 259
beta testing **393**
boundary value analysis 340, 364
branch coverage 373
bucket sizing 150
bug hunt 390
built-in quality 19, 194
business analyst 21, 35
business case 38
Business Test Stage 54
business value 45

C

CALMS 77
CC 359
CD 56
CDC 359
chance of failure **46**
chance of faults 46
charter **384**

checking 291
checklist 371, 377, **379**
CI/CD 19, 53, 78
CI/CD pipeline 53, 57, 59, 77, 169, 241
CI/CD tooling 32, 59
classification tree 368
cloud adoption 242
cloud technology **169**
cloud-based infrastructure 166
CM **57**
coach 21, 35
Code 20, 75
code coverage 328, 371
code coverage types 372
cognitive QA 128
collaboration techniques 278
combining coverage- and experience-based testing 378
completion criteria **52**
condition coverage 359
condition decision coverage 359, 373
condition-oriented test design 345, 358
confidence 17, 35, 45, 47
Confidence in pursued value 37
confidence monitor 142
configuration as code 247
confirmation test **292**
containerization **169**
Continuous Delivery 53, **56**, 58
Continuous Deployment 19, 53, **56**, 58
continuous everything 18, 55, 77
continuous improvement 99, 182, 193
Continuous Integration 19, 53, **55**, 57, 77
continuous monitoring 55, **57**, 58
Continuous Quality & Testing **57**, 58
continuous quality engineering 18
continuous test automation 238
continuous testing 19, 44, 238, 241-242, 251
continuous value range 362
control **125**
control flow testing 359
corrective quality measures 220
cost of quality 115
coverage 40
coverage group **344**
coverage level 367
coverage types for condition coverage 359
coverage-based testing 230, 270, 333, **334**, 337
CQT **57**
criteria 51
cross-functional DevOps team 119
cross-functional team 18, 29-30
crowd sourcing 390
crowd testing 377, 382, **390**
crowdsourced testing 390

D

DASA 30, 122

dashboard 47, 49, 138
dashboarding 128
data combination test 366
Data Cycle Test 396
data-oriented test design 346
data-oriented testing 362
DC 359
DCoT 366
debriefing 388
decision coverage 359, 373
decision points 358
decision table 360
decision table testing 359
defect 134
defect management 134
Definition of Done 52, 216, 301
Definition of Ready 52
demand party 86
demand/supply model 86
Deming cycle 202
Deploy 20, 37, 76
descriptive analytics 129
designer 308
detailed reporting 137
detective quality measures 220
developer 21, 31, 35
DevOps **17**, 27, 29, 53, 63, 73, 104
DevOps activities 19-20, 74, 104
DevOps Agile Skills Association 122
DevOps efficiency metrics 188
DevOps principles 30
DevOps role 38
DevOps team 18, 332
DevOps tooling 177
discrete values 362
documentation as code 248
DoD 52
DoR 52
DORA DevOps performance metrics 188
dot voting 152
drivers 56
dynamic testing **51**, 222

E

EDD 259
effectiveness **183**, 186
effectiveness metrics 186
efficiency **183**, 186
efficiency metrics 187
elementary comparison testing 359
end user 307
end-to-end performance testing 301
end-to-end test 109, 251, 292
entry criteria **52**
equivalence partitioning 362
error **133**
error guessing 377, 381
error seeding 326
estimating 99
estimating techniques 146

estimation based on ratios 153
estimation techniques 145
example mapping 262
example-driven development 259
executable requirements 259
exit criteria **52**
expected outcomes 252
expected results 375
Experience 35
Experience the real value 37
experience-based approach 383
experience-based testing 230, 270, 333, **334**, 337, 376
experience-based testing approaches 376
experiment toggle 272
explicit testing **250**
exploration **48**
exploratory testing 377, 381, **383**
exploratory testing charter 293, 392
exploring 280
extrapolation 153

F

failure **134**
fault 44-45, **133**
fault clustering 255
fault density 40
fault seeding 326
feature branch 271
feature toggle **271**
fixing phase 43
flaw **317**
forecasting 19
formal reviewing 226
Four amigos 224, 261
four-value BVA 364
free-style testing 383
frequency of use 46
functional system indicators 126

G

GDPR 231
General Data Protection Regulation 231
Generic Test Agreements 117
Gherkin 264
go/no-go decision 49
Goal-Question-Metric 41, 184, 197
GQM 41, 195, 197
GTA 117
GUI 288

H

heuristic **394**
heuristics 269, 387
heuristics for preparing dynamic tests 396
heuristics for static testing 395
high-level reporting 143
high-level reports 50
high-performance IT delivery **16**, 29, 35, 63, 70

high-performance IT delivery model 18, 26, 104
high-performance team 276
hybrid Demand/Supply model 64
hybrid IT delivery 63
hybrid IT delivery model 18, 85

I

impact 46
implicit test **250**
incident **134**
incident management 134
Indicators 35, 37, 40-41, 45-46, 125, 137, 184
indicators in VOICE model 39
informal reviewing 223
infrastructure 99, 165
infrastructure-as-code 28, 166, **167**, 247
intake test 249
Integrate 20, 75
INVEST 269
investigate and assess outcome 102
IT delivery models 18, 63
item risk **210**
iteration model 297, 304
iterative-incremental 16

K

Kanban 82
Kanban board 158
Kano model 163

L

Lean 79
learn fast 280
line coverage 373
load model 296, **302**
logical test case **340**
logical test design 341

M

machine learning 19, 32
magic boat ride 352
maintainability 321
maintainability testing 321
maintenance testing 322
manager 35
masking 233
master test plan 97
MCC 359-360
MCDC 359
metrics 99, 181
mob programming 268, 279
mob testing 387
mobbing 278
mocks 56
Model-Based Review 225
model-based testing 242
modified condition decision coverage 359, 373
Monitor 20, 36-37, 75
monitoring 43, **125**

monitoring & control 99, 124
monitoring of product quality 273
MoSCoW 160
multi-disciplinary team 29-30
multiple condition coverage 359, 373
mutation testing **326**

N

n-amigos session 224
no risk – no test – no development 352
non-functional system indicators 127
non-functional test 251
NoOps 33
n-wise testing 370

O

Objectives 35, 37, 46
observability 274
Operate 20, 37, 76
operations person 21, 35
ops toggle 273
Organizing 20
Organizing topics 94, 98
overview reporting 138
overview reports 50
OWASP top 10 318

P

pair programming 267
paired testing 386
pairing 268, 278
pairwise testing **370**
parallel testing 274
path coverage 353, 373
pen test 319
penetration test 319
people 17, 20, 26, 38
performance activities 298
performance efficiency **294**
performance metrics plan 297, 305
performance testing 294, 296
Performing 20
Performing topics 94, 101
permission toggle 273
personas 311
physical test case **341**
physical test design 341
pipeline as code 247
Plan 20, 36-37, 75
Plan, Do, Check and Act 202
planning 99, 155
planning poker 147, 207
post-industrial mindset 28
post-industrial testing 31
predictive analytics 129
prescriptive analytics 129
presentation test 371
pre-test **249**
preventive quality measures 220
prioritization approaches 158

prioritizing with MoSCoW 160
problem **130**, **134**
process cycle test 355
process-oriented test design 345
product owner 21, 35
progression test **292**
progression testing 291, 358
pseudo code 371
pull request **224**
pursued business value 350
pursued value 17, 30

Q

QA 19
QA & testing topics 94, 103
QAM 197
QPM 200
quality **42**, 44-46, 51, 138
quality & test policy 112
quality assurance 19
quality assurance & testing topics 20
quality at speed 19, 238, 245
quality characteristics 45, 209, 348
quality engineer 21
quality engineering 18, 26, 31, 98, 101, 219, 332
quality forecasting 128
quality gate 88
quality indicators 125
quality level 44-45
quality measures 102, 219, 258, 350
quality policy 112
quality risk 40, **46**, **206**, 337, 348, 351
quality risk analysis 46, 102, 206, 340, 350
Quality to Activity Mapping 197
Quality to People Mapping 200

R

red teaming **319**
refactoring 28
registering anomalies 228
regression test **292**
Regression testing 291, 358
release toggle 272
remaining risks 44
report 49
reporting 47
reporting & alerting 99, 135
requirements specification 339
respondent 308
responsibilities 118
responsibilities & roles 99
re-test **292**
reviewing 102, 222
risk points 207
risk poker 147, 206
risks 46, 139
roles 118
rolling wave planning 156
root cause analysis 258

S

SaE 259
SAFe model 89, 107-108
SbE 259
Scaled Agile Framework 64, 89
scripted testing 383
Scrum 29, 71
Scrum board 158
Scrum framework 63
Scrum guide 71, 122
security **314**
security operations center 320
security testing 314
security testing approaches 317
semantic test 370
semantic testing 359
semantic testing using MCDC 360
sequential IT delivery 63
sequential IT delivery models 28, 66, 103
sequential models 18
serious fun 390
service virtualization 167, 242
skill level 351
skills 35, 276, 349
skills matrix 281
SMART 269
smart automated frameworks 246
smoke test 249
soap opera testing 389
social engineering 319
Specification and Example 259
Specification and Example approach 259
Specification by Example 259
Specification by Example technique 262
spheres of testing 287, 292
Spotify model 82
staff organization 281
stakeholders 38, 44, 47, 49
statement coverage 373
static analysis 226
static testing **50**, 102, 222
story points 147, 207
strong-style pairing 269
structured test design 334
structured testing 230, 270, 332
stubs 56
subsetting 233-234
supply party 86
swim lane sizing 150
syntactic testing 370
synthetic data 234
synthetically generated data 233
system test 293
systems architect 35
systems thinking 30

T

Tayloristic testing 31
Tayloristic view 26, 28
TDD 265

TDL-n 356
TDM 231
team member 19
team performance indicators 126
team roles 38
Team Test Stage 54
team values 279
technical debt 321
telemetry 178, 274
test approach 337, **343**
test automation 102, 238, 243
test basis 229, **339**, 348, 351
test case **340**
test control tools 172
test coverage 328
test data 232, 340, **341**
test data management 102, 231, 233
test data management practices 236
test depth level-N **356**
test design 102, 229, **230**, 332, 338
test design entities 338
test design entities relationship diagram 338
test design technique **343**, 350, 387
test environment 165, **166**
test environment tools 172
test execution 102, **249**, 340
test execution tools 172
test idea **387**
test infrastructure **166**
test intensity 212, 343
test intensity table 349-350
test log **387**
test orchestration 244
test pass/fail ratio 40
test plan 95, 97
test policy 112
test professional 31, 122
test reports 388
test scenario 338, **342**
test script **342**
test session 378
test situation **339**
test strategy 102, **206**, 293, 343, **350**, 367, 376
test suite 56
test tool capabilities 171
test tooling 171, 173
test variety 250, **287**, 292
testability **324**
test-driven development 213, 265
test-driven requirements 259
tester 21, 35
testing **42**, 47, 49, 51
testing challenge 351
testing pyramid 239, 288-289
testing quadrants 239, 290
testing tours 389
TestLand 352
testware 49

threat modelling 226
Three amigos 261
three ways of DevOps 30
three-value BVA 364
TMAP 23
tooling 99, 171
topics 94
T-shaped 282
T-shirt sizing 149
types of test tooling 173

U

unit integration testing 292
unit testing 56, 292
unstructured testing 270, 332, 398
usability **306**
usability testing 306
User eXperience testing 307
user researcher 308
user story 216
UX test 251
UX testing 307

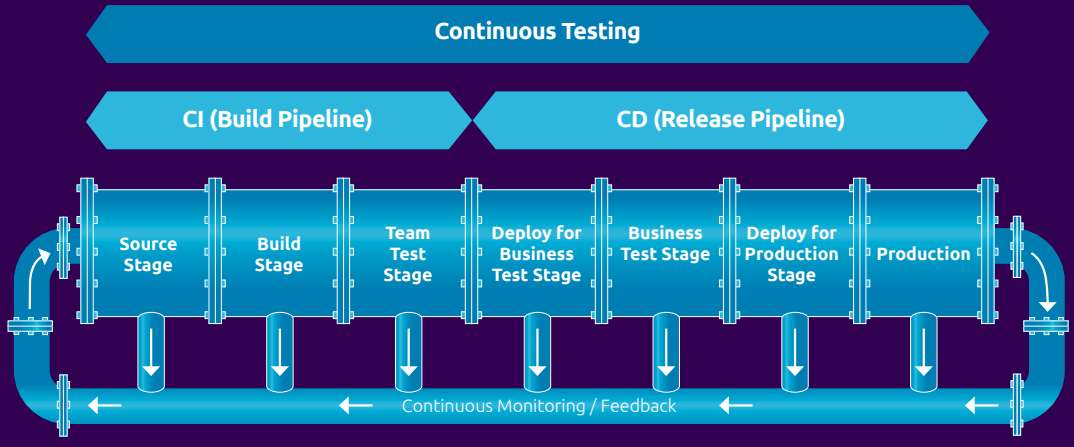
V

V model 18, 63, 68, 103
validating 291
validation **48**
validation test 370
Value 35-36
varieties of performance testing 294
Velocity 40, 140, **146**
verification **48**
verifying 291
VOICE model 21, **34**, 36-38, 47, 184
VOICE model applied to DevOps 36
vulnerability **314**

W

waterfall 18
Waterfall model 63, 66, 103
Weighted Shortest Job First 160
WSJF 160

To continuously deliver IT systems at speed with a focus on business value, DevOps teams integrate quality engineering in their way of working.
This book supports teams in implementing quality in their DevOps culture, with practical examples, useful knowledge and some theoretical background. For example, it describes how to benefit from a CI/CD pipeline.



TMAP is the body of knowledge for quality engineering in IT delivery and builds on practical experience from thousands of people in more than twenty-five years. The website, www.tmap.net, supports any kind of IT delivery model. This book, however, focuses on DevOps: today's implementation of high-performance IT delivery.

The authors of this book have combined experience in every relevant role in today's high-performance IT delivery. While creating this book, we practiced what we describe about bringing together all disciplines needed to deliver business value. This book would not have existed without the efforts of **Leo van der Aalst**. His contribution is based on three decades of IT experience combined with 21 years of TMAP knowledge. Leo wrote large parts of this book and supported the authors in many ways. Many thanks to him! Another vital source of knowledge and experience was **Richard Ammerlaan**, who wrote the parts about automation, tooling and performance testing.

sogeti
Part of Capgemini

ISBN 978-90-75414-89-9
9 789075 414899