

Managing Software Process Evolution

Marco Kuhrmann · Jürgen Münch
Ita Richardson · Andreas Rausch
He Zhang
Editors

Managing Software Process Evolution

Traditional, Agile and Beyond –
How to Handle Process Change



Springer

Editors

Marco Kuhrmann
University of Southern Denmark
Odense
Denmark

Jürgen Münch
Reutlingen University
Reutlingen
Germany

Ita Richardson
University of Limerick
Limerick
Ireland

Andreas Rausch
Clausthal University of Technology,
Department of Informatics
Clausthal-Zellerfeld
Germany

He Zhang
Nanjing University
Nanjing
China

ISBN 978-3-319-31543-0 ISBN 978-3-319-31545-4 (eBook)
DOI 10.1007/978-3-319-31545-4

Library of Congress Control Number: 2016934950

© Springer International Publishing Switzerland 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Foreword

Whether we like it or not, the business software is a fashion industry where new fads come and go. All of us can name several of them from the past years and share the war stories about attempting to fix problems with the newest fad at everyone's lips. Today some of these fads are called continuous deployment, lean startups, DevOps, flow and any of the Japanese sounding concepts from the lean vocabulary such as Kanban for some years now. So, who cares about software processes anymore?

Back in the days of my software development in the 1990s in a 10-person software boutique producing production planning software for Nokia factories, I became worried about the quality of our software and shared my thoughts with the chief architect. He made me a quality manager for the company and a few weeks later Nokia performed an ISO 9001 assessment for our boutique and the software. It turned out that we deviated in all other points but at least we had an appointed quality manager to worry about the results. Among others, we did not have processes in place.

Later I made my Ph.D. in Software Process Improvement, became a certified BOOTSTRAP and SPICE assessor, and discovered that if there is a way to improve performance, all sorts of process improvement endeavors are not the way forward. As much as we'd like, the processes are not tangible artifacts, they do not deliver software and when improved, we cannot quantify the results in order to determine whether we are developing better software than yesterday.

The past decade and a half has still been mostly about the processes, methods, and tools; but in a refreshing way. Agile kicked out process developers and empowered developers to become concerned about the business of the company, the actual software development. Agile grew rapidly outside of its origins from small teams to larger teams, to organizations and to global software development. Yes, we still agree to disagree what exactly agile is and what it means in different contexts in precise terms. We have learned that each and every organization must define it for themselves. This has proven to be painful but necessary when the ideas of continuous improvement and learning organizations are nurtured.

I am glad to say that the industry is moving forward, academics are reforming the curriculums, practitioners are more involved than ever before, and improvements or even improvement leaps are being made.

This is where the book at your hand becomes very handy. It shows how software processes evolve and are impacted in various contexts and how this can be managed. In an interesting way the book contributes to the era beyond agile, which will be welcomed by many. Also, I should note that the book does not attempt to anchor itself in one particular domain, context or business situation but offers different perspectives, which will benefit readers with different backgrounds.

The editors have collected 15 chapters from authors that represent both the academic and the practitioner standpoints. Some of the chapters are refreshingly controversial and many provide also concrete guidance on how to make a lasting impact. All of the chapters communicate the same underlying message, which I feel is very important: “We must keep changing the way the software is being developed.” I particularly appreciated the sense of urgency to push the industry forward from a multitude of different viewpoints. While all chapters have their merits, I will highlight a few that caught my attention:

Anthony I. Wasserman (Chap. 1) presents the ultimate key question that has puzzled us for a number of years: “How much process is needed by an organization for a particular project?”. In many ways he sets the stage for the remainder of the book by arguing that there are only a few places left where high-ceremony (a.k.a. traditional) processes are needed.

Andreas Rösel (Chap. 5) challenges the reader by presenting an approach to guarantee a failure when gigantic improvements should be sought but when we do not dare to take the risk. Many of the anti-tactics he identifies should be an eye-opener for software managers thinking about their improvement efforts.

Christian Prause and his colleagues’ (Chap. 8) chapter, on the other hand, is an eye-opener for all those people who think software development is an easy endeavor. They describe their environment insightfully: “Scientific missions have no insurance; a second unit is never built. If the mission goal is not reached, for whatever reason, there is no second chance.” Software that needs to operate for decades in the outer space requires all the ceremonies invented but a smart way of executing them.

Kai Petersen’s (Chap. 12) contribution should be an absolute read to all researchers and practitioners involved in academia–industry collaboration. In a very practical way, he summarizes the literature on the topic but also adds his personal advice on how the research done by the academics can deliver the maximum value for the company.

Yli-Huomo and his colleagues (Chap. 15) demonstrate how changes in the organization and the way the software is being developed do have a fairly direct impact on the technical debt that hinders the company to progress and move forward in their development. The readers will particularly enjoy the illustrative quotes from the interviews in three large companies.

Coming back to where I started, it turns out that software processes have become ever more tangible and concrete actions can be taken that will show a difference in the bottom line. The editors have put together a book that does a splendid job in fulfilling an evident gap in the current literature by shaping the state of the art in software process evolution scene.

Whether you read this book from start to finish, or piecemeal your approach iteratively, I am sure you will find this book as valuable as I did.

Trondheim, Norway
January 2016

Pekka Abrahamsson
Professor of Software Engineering
Norwegian University of Science and Technology

PS: I forgot to conclude the story about the ISO 9001 assessment in the software boutique where I worked. After surviving the shock of being evaluated by Nokia, my company jumped in the process wagon, eventually I departed to pursue academic studies, and the company ended up being recognized by International Quality Crown award in 2008 and Arch of Europe Quality award in 2010. Thus, presenting a happy ending for one process assessment case.

Preface

Imagine this happens to you: Your manager tells you “Agile is the future! Let’s go Scrum.” He *forces* you to replace the existing development process with Scrum. What would you do? Would you send your developers to a Scrum training course immediately?

It is true that more companies are embracing agile as part of their development process in order to increase speed, accelerate learning, and deliver value rapidly. And many of these companies are applying Scrum. But it is also true that evolution does not follow the principle: “Progressive dinosaurs are the future! Let’s go bird.”¹

Evolving the ways through which software-intensive products and services are developed is a challenging endeavor that needs to be done carefully. Where do you start? What do you have to consider?

This book will help you better understand the different aspects and challenges of evolving development processes. It addresses difficult problems, such as how to implement processes in highly regulated domains or where to find a suitable notation for documenting processes. This book emphasizes the need to consider *Software Process Evolution* as an important means for catching up with rapid changes in technical and market environments. It provides insights that might help you manage process evolution. It gives plenty of tips, e.g., how to cope with the threat of disruption from a process perspective. In addition, it provides many examples and cases on how to deal with software evolution in practice.

Why a Book on Managing Process Evolution?

Many organizations need to transform their business to the next level. In order to benefit from leading-edge technologies, catch up with the digital transformation, and continuously innovate and renew business models, companies have to quickly

¹Quote taken from a tweet from David Evans.

adapt and change the ways they develop products and services. As software is the key to this transformation, the ways in which modern software is developed need to change accordingly.

Another important driver for process evolution is the need to mitigate software risks. Basically, a considerable share of software risk is process-based [3]. For example, there have been several incidents which could have been avoided with appropriate coding standards and tools. Although these standards and tools are widely available, they are either not applied or not appropriately applied in many situations. Because this is normally caused by the way work and people are organized and work is carried out it is a software process issue. Companies need to find ways to ensure that process models are properly defined and, furthermore, are appropriately applied while not hindering the creativity of, e.g., designers or developers. To do this effectively, defining and deploying adequate software processes usually requires fostering the evolution of existing processes and their underlying models towards ones that suit better.

Today, there exists a variety of software processes ranging from generic and domain-specific standards, from agile methods to comprehensive process engineering frameworks. Since software processes may contain up to hundreds or even thousands of elements, the management of a software process is a demanding task and, therefore, many companies install whole departments dealing with software process improvement and management. In practice, especially in large organizations, we can observe some interesting gaps:

- Development teams tend to apply agile methods while the hosting organization focuses on “classic” structured development processes [5, 6].
- Implemented development processes in projects differ from what has been defined [4].
- Evolving software technologies and platforms require a parallel evolution of software processes to accommodate the rapid changes. However, this co-evolution does not appropriately take place.

One main reason for these gaps is different mindsets. For instance, program managers and quality assurance people need planned and directed processes for certification, budgeting, and compliance business. Developers need flexibility and processes which support creative work. Business managers need processes that allow for fast results and flexible feature delivery. Moreover, due to technology evolution, business evolves. This requires that emerging markets must be addressed, new technologies should be adopted, and globally distributed development becomes more and more important.

Apart from the big “global players,” process evolution is also highly important for small and medium-sized companies. Such companies typically neither have comprehensive process models nor process engineering groups, and often have to trust in a common understanding of principles and applied practices. However, these principles and practices need to be continuously validated against higher level goals (such as business strategies) and potentially changed in order to secure and maintain the company’s position in the market place [2]. One example for such a

change is the increasing focus on value-delivery [1]. Regardless of the company size, a major challenge that companies face is to provide all stakeholders with flexible processes that:

- Are driven by the needs of the different stakeholders,
- Have clear links to higher level goals of an organization,
- Provide interfaces that are compatible with organizational structures,
- Are supported by tools for modeling, enactment, analyses, and evolution,
- Can be tailored to individual project goals and characteristics,
- Offer adaptability and elasticity to accommodate and support technological and organizational innovations and evolutions.

This book focuses on the design, development, management, governance, and application of evolving software processes that are aligned with changing business objectives, such as expansion to new domains or moving to global production. In the context of evolving business, it addresses the complete software process life-cycle, from initial definition of a product to systematic improvement.

Who Should Read This Book?

This book is aimed at anyone interested in understanding and organizing software development tasks in an organization. The experiences and ideas in this book are useful for both those who are unfamiliar with software process improvement and want to get an overview of the different aspects of the topic, and those experts with many years of experience. In particular, the present book addresses researchers and Ph.D. students in the area of Software & Systems Engineering and Information Systems, who study advanced topics of organizing and managing (software development) projects and process improvements projects. Furthermore, the book addresses practitioners, consultants, and coaches involved in software-related change management and software process improvement projects, and who want to learn about challenges and state-of-the-art techniques and experiences regarding their application to problems in different application domains.

How is the Book Organized?

This book is organized in three parts (Fig. 1). Part 1 focuses on software business transformation, its challenges, and addresses the questions about which process(es) to use and adapt, and how to organize process improvement programs. In Chap. 1, Tony Wasserman discusses short lifecycle projects and how “low-ceremony processes” help shorting project iterations. In this context, in Chap. 2, Diebold and Zehler discuss the “right” degree of agility in rich software processes—how to find and how to achieve this. The challenge of implementing agile software

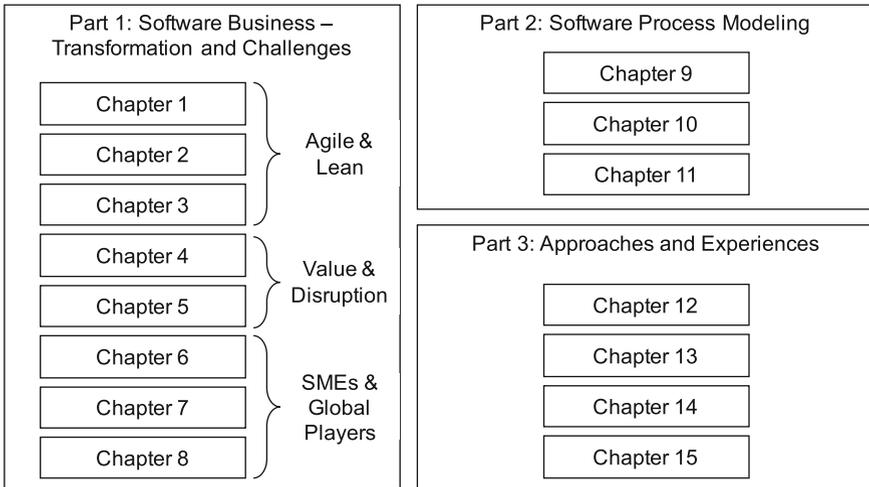


Fig. 1 Overview of the book and chapter outline

development approaches is further discussed by Houston and Rosemergy in Chap. 3, who report an agile transformation of a globally distributed company. As many companies jump to Agile processes hoping for the benefits promised, determining value and value creation is crucial. In Chap. 4, Christof Ebert discusses the principles of value-driven process management and reports experiences. Another perspective is taken by Andreas Rösel, who describes how concepts of design thinking can be applied to disruptive improvements in Chap. 5. Oisín Cawley discusses the trials and tribulations of Global Software Engineering processes in the course of business evolution with a particular focus on regulated software and system development in Chap. 6. The respective Software Process Improvement challenges, approaches, and standards for very small entities and small- to medium-sized companies are presented in Chap. 7 by Mary-Luz Sánchez-Gordón and her colleagues. In their systematic literature review, they give a comprehensive overview of the different improvement approaches and models and show how they find their way into international standards. Standards and their role are also key to the Space business, as presented in Chap. 8, where Christian Prause and his colleagues describe how software processes in the German Space Administration evolve and how they are tailored to the projects.

Part 2 of the book is focused on process modeling. This part starts with Chap. 9 by Dumas and Pfahl, who discuss the appropriateness of the Business Process Model and Notation (BPMN) for software processes modeling. In Chap. 10, Fazal-Baqaie and Engels present an approach to modeling evolving software processes by utilizing method engineering principles. The adaptation of case management techniques for the purpose of improving process model flexibility is demonstrated by Marian Benner-Wickner as his colleagues in Chap. 11.

Finally, Part 3 of the book collects approaches, experiences, and recommendations that help to improve software processes with a particular focus on specific lifecycle phases. The part starts with Chap. 12 in which Kai Petersen reports his experiences in industrial Software Process Improvement projects from the perspective of a researcher. He reports from projects and provides a collection of general lessons learned and recommendations to aid researchers and practitioners to plan and carry out improvement projects in an industry–academia collaboration. Chapter 13 in which Regina Hebig and her colleagues give insights into two large-scale industry projects and demonstrate how co-evolution is manifested and handled in such projects, thus addressing the co-evolution of software processes and model-driven engineering approaches. In Chap. 14, S.M. Didar Al Alam and his colleagues present an approach that helps companies to improve the release readiness of their software products. They show how bottleneck factors that hinder fast releases can be detected and they apply their concept to different Open-Source Software projects. Finally, Jesse Yli-Huumo and colleagues take a broader perspective in Chap. 15, discussing how process evolution affects technical debt. They illustrate their findings with three large-scale software projects.

We wish you an interesting and enjoyable reading experience. A collection such as this book would not be possible without the help of many persons. We would especially like to thank the authors for their insightful articles and their excellent collaboration. In addition, we would like to thank Ralf Gerstner from Springer, who supported us efficiently in completing organizational and contract issues.

Odense, Denmark
 Reutlingen, Germany
 Limerick, Ireland
 Clausthal-Zellerfeld, Germany
 Nanjing, China
 January 2016

Marco Kuhrmann
 Jürgen Münch
 Ita Richardson
 Andreas Rausch
 He Zhang

References

1. Bosch, J.: Speed, data, and ecosystems: The future of software engineering. *IEEE Softw.* **33**(1), 82–88 (2016)
2. Fagerholm, F., Guinea, A.S., Mäenpää, H., Münch, J.: Building blocks for continuous experimentation. In: *Proceedings of the International Workshop on Rapid Continuous Software Engineering, RCoSE*, pp. 26–35. ACM, New York (2014)
3. Neumann, P.G., et al.: Column: Risks to the Public. *ACM SIGSOFT Softw. Eng. Note* **40**(6), 14–19 (2015)
4. Parnas, D.L., Clements, P.C.: A Rational Design Process: How and Why to fake it. *IEEE Trans. Software Eng.* **12**(2), 251–257 (1986)

5. Theocharis, G., Kuhmann, M., Münch, J., Diebold, P.: Is Water-Scrum-Fall reality? On the use of agile and traditional development practices. In: Proceedings of the International Conference on Product-Focused Software Process Improvement. Lecture Notes in Computer Science, vol. 9459, pp. 149–166. Springer, Heidelberg (2015)
6. Vijayarathy, L., Butler, C.: Choice of software development methodologies - do project, team and organizational characteristics matter? *IEEE Software* (99), 1ff. (2015)

Contents

1	Low Ceremony Processes for Short Lifecycle Projects	1
	Anthony I. Wasserman	
2	The Right Degree of Agility in Rich Processes	15
	Philipp Diebold and Thomas Zehler	
3	Assessing Product Development Agility	39
	Daniel X. Houston and Stephen W. Rosemergy	
4	Value-Driven Process Management	61
	Christof Ebert	
5	Are We Ready for Disruptive Improvement?	77
	Andreas Rösel	
6	Trials and Tribulations of the Global Software Engineering Process: Evolving with Your Organisation	93
	Oisín Cawley	
7	The Route to Software Process Improvement in Small- and Medium-Sized Enterprises	109
	Mary-Luz Sánchez-Gordón, Ricardo Colomo-Palacios, Antonio de Amescua Seco and Rory V. O'Connor	
8	Managing Software Process Evolution for Spacecraft from a Customer's Perspective	137
	Christian R. Prause, Markus Bibus, Carsten Dietrich and Wolfgang Jobi	
9	Modeling Software Processes Using BPMN: When and When Not?	165
	Marlon Dumas and Dietmar Pfahl	

10 Software Processes Management by Method Engineering with MESP 185
Masud Fazal-Baqaie and Gregor Engels

11 Adapting Case Management Techniques to Achieve Software Process Flexibility 211
Marian Benner-Wickner, Matthias Book and Volker Gruhn

12 A Researcher’s Experiences in Supporting Industrial Software Process Improvement. 235
Kai Petersen

13 Lessons Learned from Co-Evolution of Software Process and Model-Driven Engineering. 257
Regina Hebig, Andreas I. Schmied and Ingo Weisemöller

14 Monitoring and Controlling Release Readiness by Learning Across Projects. 281
S.M. Didar Al Alam, Dietmar Pfahl and Günther Ruhe

15 The Effects of Software Process Evolution to Technical Debt—Perceptions from Three Large Software Projects. 305
Jesse Yli-Huumo, Andrey Maglyas and Kari Smolander

Index 329

Contributors

Marian Benner-Wickner is Research Associate at the University of Duisburg-Essen and a professional IT Staff Trainer at CampusLab. His research interest includes software support for case management. After completing a vocational training as an IT specialist (2005), he studied computer engineering at the University of Applied Sciences in Dortmund. During his course of study, he worked on a research project at Fraunhofer ISST. After receiving his M.Sc. in 2010, he started his Ph.D. studies under the guidance of Prof. Volker Gruhn. Contact Marian at:

paluno—The Ruhr Institute for Software Technology, University of Duisburg-Essen, Germany.

E-mail: marian.benner-wickner@uni-due.de

Markus Bibus is Product Assurance Manager at DLR Space Administration for 6 years. He is the responsible product assurance manager in several DLR Space Administration projects such as Laser Communication Terminal, eRosita, ExoMars PANCAM High Resolution Camera, and CAPTARE with focus on EEE parts including FPGA programming, materials and processes. His background includes physics (Diplom, equiv. to M.Sc.), specialized in semiconductor technology and computer science as minor subject. Prior to his current tasks he worked as quality manager and technology engineer in semiconductor manufacturing business, in both research & development and automotive branches. Contact Markus at:

Deutsches Zentrum für Luft- und Raumfahrt, DLR—Space Administration, Königswinterer Straße 522-524, Bonn, Germany.

E-mail: markus.bibus@dlr.de

Matthias Book is Associate Professor for Software Engineering at the University of Iceland. After receiving his doctoral degree from the University of Leipzig, he worked as Research Manager for the German software company adesso, led the Mobile Interaction group at the University of Duisburg-Essen's Ruhr Institute for Software Technology (paluno), and served as acting head of the Software Engineering and Information Systems Chair at Chemnitz University of Technology.

His research focus is on facilitating communication between business and technical stakeholders in large software projects, and on interacting with software systems through novel input modalities such as gestures and voice commands. Contact Matthias at:

Department of Computer Science, University of Iceland.

E-mail: book@hi.is

Oisín Cawley is Lecturer and Researcher in Computing at the Institute of Technology Carlow, Ireland. He worked for 17 years in software development, predominantly for multinational companies, and has held several senior positions. He holds a B.Sc. in Computer Science, an MBA, and a Ph.D. in Software Engineering. Some of his research interests include software development processes and methodologies, particularly within regulated environments. He has a keen interest in improving the learning process for third-level computing students, and introducing young students to computing. He volunteers in his local primary school where he teaches programming to the sixth class children. Contact Oisín at: Department of Computing, Institute of Technology Carlow, Ireland.

E-mail: oisin.cawley@itcarlow.ie

Ricardo Colomo-Palacios is Professor at the Computer Science Department of the Østfold University College, Norway. Formerly he worked at Universidad Carlos III de Madrid, Spain. His research interests include applied research in information systems, software project management, people in software projects, business software, and software and services process improvement. He received his Ph.D. in Computer Science from the Universidad Politécnica of Madrid (2005). He also holds an MBA from the Instituto de Empresa (2002). He has been working as software engineer, project manager, and software engineering consultant in several companies including Spanish IT leader INDRA. Contact Ricardo at:

Østfold University College, Halden, Norway.

E-mail: ricardo.colomo@hiof.no

Antonio de Amescua Seco has Ph.D. in Computer Science and is Full Professor in the Computer Science Department at Carlos III University of Madrid with more than 30 years of experience in higher education. His main research areas are software architect, project management, process improvement, change management, MOOC. He was the research project leader for the development of the standard of Information System Development Methodology for the Spanish Administration and has participated in projects sponsored by the European Union. He is member of the Chair of the SPI for Ibero-American Space and also member of European Certification and Qualification Association (ECQA). Contact Antonio at: Universidad Carlos III de Madrid, Madrid, Spain.

E-mail: amescua@inf.uc3m.es

S.M. Didar Al Alam is a doctoral candidate in the University of Calgary's Software Engineering Decision Support Lab. He is also a Lecturer in Computer Science Department, Islamic University of Technology. His main research interests

are readiness measurement of software release, planning of software release, decision support, data analytics, and empirical software engineering. Didar received his Master's and Bachelor's degrees in Computer Science and Information Technology from Islamic University of Technology. He held a doctoral scholarship from Alberta Innovates of Technology Futures (AITF) for more than 3 years. Contact Didar at:

University of Calgary, Department of Computer Science, Calgary, AB, Canada.

E-mail: smdalam@ucalgary.ca

Philipp Diebold is working as a researcher in the Process Engineering Department of the Fraunhofer Institute for Experimental Software Engineering IESE in Kaiserslautern (Germany) for some years. His technical focus is mainly on improving software and system development processes with the use of agile development methods such as Scrum or XP. In addition to his Fraunhofer work, he is doing his Ph. D. in the University of Kaiserslautern on introducing more elements from agile development into regulated environments, such as automotive, avionics, healthcare, by integrating agile practices in existing rich processes. Contact Philipp at:

Fraunhofer Institute for Experimental Software Engineering (IESE), Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany.

E-mail: philipp.diebold@iese.fraunhofer.de

Carsten Dietrich is a staff member of the German Aerospace Centre (DLR) in Space Administration. He had set up and implemented the customer-side software product assurance in space projects. After harmonization of ECSS system between software quality assurance and engineering, he is now responsible for the support of software engineering in space projects. Carsten Dietrich holds a diploma in Information Technology (Computer Science) from the Technical University of Braunschweig in Germany. Prior to working for DLR, Mr. Dietrich worked in several other fields, such as project management, system analysis, and quality assurance in national and international automotive and railway–industry projects. Contact Carsten at:

Deutsches Zentrum für Luft- und Raumfahrt, DLR—Space Administration, Königswinterer Straße 522-524, Bonn, Germany.

E-mail: carsten.dietrich@dlr.de

Marlon Dumas is Professor of Software Engineering at the University of Tartu, Estonia, and Adjunct Professor of Information Systems at Queensland University of Technology, Australia. His research interests span across the fields of software engineering, information systems, and business process management. His ongoing work focuses on combining data mining and formal methods for analysis and monitoring of business processes. He has published extensively in conferences and journals across the fields of software engineering and information systems and has co-authored two textbooks in the field of business process management. Contact Marlon at:

University of Tartu, Institute of Computer Science, Tartu, Estonia.

E-mail: marlon.dumas@ut.ee

Christof Ebert is Managing Director at Vector Consulting Services. He supports clients around the world to sustainably improve product strategy and product development and to manage organizational changes. Dr. Ebert serves on advisory and industry bodies and is Professor at the Universities of Stuttgart and Paris. Contact Christof at:

Vector Consulting Services, Ingersheimer Straße 24, 70499 Stuttgart, Germany.
E-mail: Christof.Ebert@vector.com

Gregor Engels received his Ph.D. in Computer Science in 1986 from the University of Osnabrück, Germany. Between 1991 and 1997, he held the position as Chair of Software Engineering and Information Systems at the University of Leiden, The Netherlands. Since 1997, he has been Professor of Informatics at the University of Paderborn, Germany. Currently, he is also director of two technology transfer labs at the University of Paderborn, the C-LAB, a joint venture together with ATOS, and the s-lab—Software Quality Lab. His research interests are model-driven software development, software architecture, and software quality assurance. Contact Gregor at:

University of Paderborn, Database and Information Systems Research Group, Zukunftsmeile 1, 33102 Paderborn.
E-mail: engels@uni-paderborn.de

Masud Fazal-Baqaie studied Computer Science at the Paderborn University with stays at the Carleton University in Ottawa, Canada and the lab IBM Research—Zurich in Switzerland. He is member of the Database and Information Systems Research Group led by Prof. Dr. Gregor Engels and consultant at the s-lab—Software Quality Lab. He is also the vice chairman of the special interest group on process models for business application development at the German Informatics Society (GI). His research interests are provisioning of organization- and project-specific software development processes, global software development, and requirements engineering and management. Contact Masud at:

University of Paderborn, Database and Information Systems Research Group, Zukunftsmeile 1, 33102 Paderborn.
E-mail: masudf@uni-paderborn.de

Volker Gruhn holds the chair for Software Engineering at the University of Duisburg-Essen. His research interests are mobile applications and software processes. Before that he held the chair for Applied Telematics and e-Business at the University of Leipzig. He received a diploma degree (1987) and a Ph.D. (1991) both in Computer Science from the University of Dortmund. Volker Gruhn is author and co-author of about 270 journal and conference articles. He co-founded the software company adesso in 1997, currently deploying more than 1700 people. Volker Gruhn was program chair of the ESEC Conference (in 2001), the German Software Engineering Conference (in 2006) and program co-chair of the ICSE Conference 2008. Contact Volker at:

paluno—The Ruhr Institute for Software Technology, University of Duisburg-Essen, Germany.

E-mail: volker.gruhn@uni-due.de

Regina Hebig is Assistant Professor at Chalmers University of Technology and the Gothenburg University in Sweden since summer 2015. Her research focusses on model-driven engineering, software processes, and quantitative measurement of software size and quality. From 2014 to summer 2015, Regina worked in a French and a European project as a postdoctoral researcher at the University of Pierre and Marie Curie in Paris, France. She received her doctoral degree from the University of Potsdam in 2014 for her research on the evolution of model-driven engineering in practice. Contact Regina at:

Software Engineering Division, Chalmers University of Technology & University of Gothenburg, Sweden. E-mail: hebig@chalmers.se

Daniel X. Houston is a Senior Project Leader at The Aerospace Corporation. His work is applying quantitative methods, particularly using statistics and simulation, to software engineering. His industrial background includes software development, Six Sigma Black Belt, and software measurement. He received M.S. and Ph.D. degrees in Industrial Engineering from Arizona State University. His publications include works on statistical modeling and simulation of software development processes, software process improvement, and the management of software projects, particularly the aspects of risk, product quality, and economics. Contact Dan at:

The Aerospace Corporation, Los Angeles, CA 90009-2957.

E-mail: dan.houston@aero.org

Wolfgang Jobi headed the Product Assurance Department of the DLR Space Administration for over 15 years, and in total has worked there for almost 30 years since the German space agency's foundation in the late 1980s. Even before the ECSS was founded, he drafted the first product assurance standards for use in the agency's missions and is the founder of the computer-aided tailoring methodology for quality and product assurance requirements for space applications. Wolfgang Jobi is a state-certified technician of electronics and holds a diploma in Electrical Engineering from the University of Cologne, Germany. Contact Wolfgang at: Deutsches Zentrum für Luft- und Raumfahrt, DLR—Space Administration, Königswinterer Straße 522-524, Bonn, Germany, (via Christian Prause).

Andrey Maglyas is a postdoctoral researcher in the Department of Innovation and Software at Lappeenranta University of Technology, Finland. His research interests include software product management, process improvements, and management methodologies. Maglyas has a D.Sc. (Tech) in Software Engineering from Lappeenranta University of Technology and a M.Sc. (Tech) in Management of Information Systems and Resources from Saint-Petersburg State Electrotechnical University, Russia. Contact Andrey at:

School of Business and Management, Innovation & Software, Lappeenranta University of Technology, P.O.Box 20, 53851 Lappeenranta, Finland.
E-mail: Andrey.Maglyas@lut.fi

Rory V. O'Connor is Associate Professor of Computing and the current Head of the School of Computing at Dublin City University, Ireland where he lectures in Software Engineering. He is a Senior Researcher with Lero, The Irish Software Research Centre. He is also Ireland's Head of Delegation to ISO/IEC JCT1/SC7 and editor of standard ISO/IEC 29110 part 2. In addition, Prof. O'Connor serves as the Editor in Chief of the Elsevier Journal Computer Standards and Interfaces. His research interests are centered on the processes where software-intensive systems are designed, implemented, and managed. His website address is www.roryoconnor.com, or Contact him at: Dublin City University, Dublin, Ireland.
E-mail: rory.oconnor@dcu.ie

Kai Petersen is Associate Professor at Blekinge Institute of Technology (BTH). His research focuses on software processes, software metrics, lean and agile software development, quality assurance, and software security in close collaboration with industry partners. Kai was ranked among the 18 most productive scholars in lean and agile software development based on publications in journals of a total of 448 researchers by Sun et al.: Assessment of institutions, scholars, and contributions on agile software development (2001–2012) in Journal of Systems and Software. Kai has authored over 70 publications in peer-reviewed international journals, conferences, and books. Contact Kai at: Blekinge Institute of Technology, Sweden.
E-mail: kai.petersen@bth.se

Dietmar Pfahl is Associate Professor of Software Engineering at the University of Tartu, Estonia, and Adjunct Professor of Software Engineering at the University of Calgary, Canada. His research interests include data-driven product and process analysis, management, and improvement. His work involves the application of data mining and machine learning techniques to build decision support models. He has 100+ refereed publications in software engineering conferences and journals. He is a Senior Member of both ACM and IEEE. Contact Dietmar at: University of Tartu, Institute of Computer Science, Tartu, Estonia.
E-mail: dietmar.pfahl@ut.ee

Christian R. Prause is the head of the software product assurance field in the DLR Space Administration and is responsible for software product assurance in all major projects. For advancing the field and its application in projects, he leads dedicated improvement projects and participates in ECSS standardization committees. Before joining DLR in 2012, he was a developer, a software quality manager, and a project manager at Fraunhofer FIT. He graduated (Dipl.-Inform.) from the University of

Bonn, and received his Ph.D. (Dr. rer. nat.) in Computer Science from RWTH Aachen University. Contact Christian at:

Deutsches Zentrum für Luft- und Raumfahrt, DLR—Space Administration, Königswinterer Straße 522-524, Bonn, Germany.

E-mail: christian.prause@dlr.de

Andreas Rösel is Principal Consultant with SAP since 2011, and since 2014 he has been IT Process Officer. His experience of some 30 years also includes being Principal Consultant for DNV IT Global Services & Q-Labs, Software Technology and SEPG Leader at Heidelberg Printing Machines, Department Head of the Advanced Software Center at ABB and beforehand as SPI consultant, Software Architect, Software Engineer and Lecturer in Germany and Australia. Andreas holds an MSc in Software Engineering and an Engineering Degree in Electronics. He has a keen interest in combining innovation, agility and processes and has published and presented at conferences including ESEPG, SPICE-Days, Object World, Comdex, BITKOM. Contact Andreas at:

SAP AG, Walldorf, Germany.

E-mail: andreas.roesel@sap.com

Stephen W. Rosemergy is Software Architect and Software Engineering Practice Expert at The Aerospace Corporation. Steve has more than two decades of experience in developing software-intensive systems for both commercial and nonprofit sectors and has been a practitioner of agile methods since 2005. Steve is a graduate of both the School of Computer Science and the Heinz School of Information Systems at Carnegie Mellon University. Contact Steve at:

The Aerospace Corporation, Los Angeles, CA 90009-2957.

E-mail: steven.rosemergy@aero.org

Günther Ruhe is the Industrial Research Chair in Software Engineering at the University of Calgary. His research focuses on product release planning, software project management, decision support, data analytics, empirical software engineering, and search-based software engineering. Ruhe received a habilitation in computer science from the University of Kaiserslautern. Since 2016, he is the Editor in Chief of the Elsevier journal Information and Software Technology. He is a Senior Member of IEEE and a member of ACM. Contact Günther at:

University of Calgary, Department of Computer Science, Calgary, AB, Canada.

E-mail: ruhe@ucalgary.ca

Mary-Luz Sánchez-Gordón is a Ph.D. student in Information Science and Technology at Universidad Carlos III de Madrid, Spain. She holds a Master's degree in Information Science and Technology from the same university. She studied computer engineering at Universidad Central del Ecuador, Quito, Ecuador. She also got her Master's degree in Education at this university. She has more than 10 years of experience in the software industry and 5 years in research and teaching in Ecuador. Her research interests are software process, software process improvement, and knowledge management. Contact Mary-Luz at:

Universidad Carlos III de Madrid, Madrid, Spain.

E-mail: mary_sanchezg@hotmail.com

Andreas I. Schmied is Managing Solution Architect at Capgemini since 2008. He has been involved in various IT projects along the development, production, and sales processes within the automotive industry. His work leads him into various roles, ranging from consultancy and development to engagement management, with a special interest in model-driven architectures that resonate with multi-cultural specifics. Andreas received a Ph.D. in Computer Science from Ulm University, Germany, for his work on the composition of software transformation processes. Contact Andreas at:

Capgemini Deutschland GmbH, Löffelstraße 46, 70597 Stuttgart, Germany.

E-mail: andreas.schmied@capgemini.com

Kari Smolander is Professor of Software Engineering in Department of Computer Science, Aalto University, Finland. His current research interests are software development practices and include especially the ongoing change in software and systems development practices and software development organizations related to digitalization. Smolander has a Ph.D. (2003) in Computer Science from Lappeenranta University of Technology, Finland. Contact Kari at:

Department of Computer Science, Aalto University, P.O.Box 15400, FI-00076 Aalto, Finland.

E-mail: kari.smolander@aalto.fi

Anthony I. Wasserman is Professor of Software Management Practice at Carnegie Mellon Silicon Valley, and the Executive Director of its Center for Open Source Investigation (COSI), focused on evaluation and adoption of open-source software. Earlier in his career, he was Professor of Medical Information Science at the University of California, San Francisco. He then started Interactive Development Environments (IDE), and served as its CEO for 10 years. He subsequently managed software and product development groups for several small companies before returning to academia in 2005. Tony is a Fellow of the ACM, a Life Fellow of the IEEE, and a Board member of the Open Source Initiative. He is a graduate of the University of California, Berkeley, and earned his Ph.D. in Computer Sciences from the University of Wisconsin, Madison. Contact Tony at: Integrated Innovation Institute, Carnegie Mellon University, Silicon Valley, Moffett Field, CA 94035, USA.

E-mail: tonyw@sv.cmu.edu

Ingo Weisemöller worked as a scientific employee at the TU Darmstadt and the RWTH Aachen from 2006 to 2011, focusing on model-based and generative software development, domain-specific languages, and model transformations. In 2012 he graduated with his dissertation thesis “Generation of Domain Specific Transformation Languages.” Since 2012, he is working as Software and Systems Designer at the Carmeq GmbH, Berlin. His activities at Carmeq include development, operation, and maintenance of tools for model-based software engineering,

primarily in the application and development of AUTOSAR, as well as software architectures, processes, methods, and data formats in automotive software development. Contact Ingo at:

Carmeq GmbH, Carnotstr. 4, 10587 Berlin, Germany.

E-mail: ingo.weisemoeller@carmeq.com

Jesse Yli-Huumo is a Ph.D. student in the Department of Innovation and Software at Lappeenranta University of Technology, Finland. His research interests include technical debt, process improvements, and software development methodologies. Yli-Huumo has an M.Sc. (Tech) in Software Engineering from Lappeenranta University of Technology. Contact Jesse at:

School of Business and Management, Innovation & Software, Lappeenranta University of Technology, P.O.Box 20, 53851 Lappeenranta, Finland.

E-mail: Jesse.Yli-Huumo@lut.fi

Thomas Zehler is working as a researcher in the Process Engineering Department of the Fraunhofer Institute for Experimental Software Engineering IESE in Kaiserslautern (Germany) for several years. His technical focus is mainly on software and system process improvement of development processes, especially using common best-practice models, such as CMMI or SPICE. His favorite domain is automotive, and he is a certified Automotive SPICE Provisional Assessor. Contact Thomas at:

Fraunhofer Institute for Experimental Software Engineering (IESE), Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany.

E-mail: thomas.zehler@iese.fraunhofer.de

Disclaimer

Any of the trademarks, service marks, collective marks, registered names, or similar rights that are used or cited in the book are the property of the respective owners. Their use here does not imply that they can be used for any purpose other than for the informational use as contemplated in this book.

The following table summarizes the trademarks used in this book. Rather than indicating every occurrence of a trademarked name as such, this report uses the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Automotive SPICE®		Verband der Automobilindustrie e.V. (VDA)
BPMN™	Business Process Model and Notation™	Object Management Group®
CMM®	Capability Maturity Model	Software Engineering Institute (SEI)
CMMI®	Capability Maturity Model Integration	Software Engineering Institute (SEI)
IDEAL™	The IDEAL™ Model	Software Engineering Institute (SEI)
MS Office®	MS Word®, MS Excel®, and MS PowerPoint®	Microsoft® Corporation
PSP™	The Personal Software Process™	Software Engineering Institute (SEI)
SAP HANA®		SAP SE
SPEM™	Software & Systems Process Engineering Metamodel™	Object Management Group®
TSP™	The Team Software Process™	Software Engineering Institute (SEI)
UML®	Unified Modeling Language®	Object Management Group®
V-Modell® XT		Federal Republic of Germany
