

Handbook of Software Engineering

Sungdeok Cha • Richard N. Taylor • Kyochul Kang
Editors

Handbook of Software Engineering

 Springer

Editors

Sungdeok Cha
College of Informatics
Korea University
Seoul, Korea (Republic of)

Richard N. Taylor
University of California, Irvine
CA, USA

Kyochul Kang
Professor Emeritus
POSTECH
Pohang, Korea (Republic of)

ISBN 978-3-030-00261-9 ISBN 978-3-030-00262-6 (eBook)
<https://doi.org/10.1007/978-3-030-00262-6>

Library of Congress Control Number: 2018960866

© Springer Nature Switzerland AG 2019

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. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

2019—the year of this handbook’s publication—marks the 50th anniversary of the traditional birth of the discipline of software engineering. Now with substantial maturity, software engineering has evolved from narrow concerns in “coding” to cover a broad spectrum of core areas, while mingling at the edges with the disciplines of human–computer interaction, programming language design, networking, theory, and more. This volume provides a concise and authoritative survey of the state of the art in software engineering, delineating its key aspects and serving as a guide for practitioners and researchers alike.

This handbook is unique among other handbooks (e.g., the Software Engineering Body of Knowledge or SWEBOK) in several aspects.

First, each chapter provides an organized tour of a critical subject in software engineering. The central concepts and terminology of each subject are laid out and their development is traced from the seminal works in the field. Critical readings for those seeking deeper understanding are highlighted. Relationships between key concepts are discussed and the current state of the art made plain. These presentations are structured to meet the needs of those new to the topic as well as to the expert.

Second, each chapter includes an in-depth discussion of some of the field’s most important and challenging research issues. Chosen by the respective subject matter experts, these topics are critical emphases, open problems whose solutions may require work over the next 10–15 years.

Articles in the handbook are appropriate to serve as readings for graduate-level classes on software engineering. Just as well, chapters that describe some of the most fundamental aspects of software development (e.g., software processes, requirements engineering, software architecture and design, software testing) could be selectively used in undergraduate software engineering classes.

A distinguishing characteristic of this volume is that in addition to “classical” software engineering topics, emerging and interdisciplinary topics in software engineering are included. Examples include coordination technologies, self-adaptive systems, security and software engineering, and software engineering in the cloud.

Software engineering practitioners in the field can thus get a quick but in-depth introduction to some of the most important topics in software engineering, as well as topics of emerging importance. Selective references at the end of each chapter guide readers to papers to obtain more detailed coverage on specific concepts and techniques.

No handbook can be considered complete nor will it remain relevant indefinitely due to rapid advances in software engineering technologies. Some topics are omitted here because, while having deep roots in software engineering, due to their maturity they are no longer broadly active research areas. Configuration management is an example. Some topics are, unfortunately, left out because of practical constraints. The editors believe that this handbook will best serve the community of software engineering researchers and practitioners alike if it is updated regularly.

Enjoy reading the 2019 state-of-the-art survey in software engineering presented by respected authorities in each of the subject areas. Needless to say, contributors to various chapters deserve the most credit for their generosity to share their expertise with the community and donate their precious time.

Seoul, Korea
Irvine, CA, USA
Pohang, Korea

Sungdeok Cha
Richard N. Taylor
Kyo-chul Kang

Acknowledgment

The editors of this handbook would like to thank the authors who spent many hours of their highly demanding schedule in writing the manuscripts. Without their enthusiastic support, this book would not exist. The Springer publication team, especially Ralf Gerstner, deserves a special note of appreciation for support and patience during the period when progress on this handbook project slowed.

Several people helped the editors as anonymous reviewers of various chapters, and we acknowledge their contribution. Special thanks are due, not in particular order, to: Kenji Tei (National Institute of Informatics, Japan), Kyungmin Bae (POSTECH, Korea), Moozoo Kim (KAIST, Korea), and Jaejoon Lee (Lancaster University, UK).

Finally, the editors would like to express personal acknowledgment for the support they received while working on this book project.

Sungdeok (Steve) Cha thanks his wife, Christine Yoondeok Cha, for her endless love and support in prayer. This book is my gift to you, Yoondeok. Richard Taylor thanks his wife, Lily May Taylor, for her continuing support, in this year of our 40th wedding anniversary. Kyochul Kang thanks the late Prof. Daniel Teichroew for leading him into software engineering research. Also, sincere gratitude goes to his wife Soonok Park for her loving care and support whenever he challenged for a new career.

Contents

Process and Workflow	1
Leon J. Osterweil, Volker Gruhn, and Nils Schwenzfeier	
1 Background, Goals, and Motivation	1
2 History and Seminal Work	5
3 Some Definitions, Unifying Assumptions, and Characterizations	9
4 Conceptual Framework	11
5 Specific Processes, Frameworks, and Architectures	23
6 Future Directions	35
7 Conclusions	46
References	47
Requirements Engineering	51
Amel Bennaceur, Thein Than Tun, Yijun Yu, and Bashar Nuseibeh	
1 Introduction	51
2 Concepts and Principles	53
3 Organised Tour: Genealogy and Seminal Works	59
4 Future Challenges	83
5 Conclusion	86
References	86
Software Architecture and Design	93
Richard N. Taylor	
1 Introduction	93
2 An Organized Tour: Genealogy and Seminal Papers	95
3 Concepts and Principles: Summarizing the Key Points	108
4 Future Directions	117
5 Conclusions	119
References	120
Software Testing	123
Gordon Fraser and José Miguel Rojas	
1 Introduction	124
2 Concepts and Principles	125

3	Organized Tour: Genealogy and Seminal Works	130
4	Future Challenges	178
5	Conclusions	184
	References	185
	Formal Methods	193
	Doron A. Peled	
1	Introduction	193
2	Historical Perspective	195
3	Grand Tour of Formal Methods	199
4	Future Challenges	217
	References	219
	Software Evolution	223
	Miryung Kim, Na Meng, and Tianyi Zhang	
1	Introduction	223
2	Concepts and Principles	225
3	An Organized Tour of Seminal Papers: Applying Changes	227
4	An Organized Tour of Seminal Papers: Inspecting Changes	247
5	An Organized Tour of Seminal Papers: Change Validation	262
6	Future Directions and Open Problems	268
	References	273
	Empirical Software Engineering	285
	Yann-Gaël Guéhéneuc and Foutse Khomh	
1	Introduction	286
2	Concepts and Principles	287
3	Genealogy and Seminal Papers	294
4	Challenges	303
5	Future Directions	312
6	Conclusion	316
	References	316
	Software Reuse and Product Line Engineering	321
	Eduardo Santana de Almeida	
1	Introduction	321
2	Concepts and Principles	322
3	Organized Tour: Genealogy and Seminal Papers	329
4	Software Product Lines (SPL): An Effective Reuse Approach	341
5	Conclusion	345
	References	346
	Key Software Engineering Paradigms and Modeling Methods	349
	Tetsuo Tamai	
1	Introduction	349
2	Organized Tour: Genealogy and Seminal Works	350
3	Future Challenges	368

4	Conclusions	371
	References	372
	Coordination Technologies	375
	Anita Sarma	
1	Introduction	375
2	Organized Tour of Coordination Technologies	377
3	The Coordination Pyramid	378
4	Conclusion and Future Work	391
	References	394
	Software Engineering of Self-adaptive Systems	399
	Danny Weyns	
1	Introduction	399
2	Concepts and Principles	401
3	An Organised Tour in Six Waves	406
4	Future Challenges	432
5	Conclusions	439
	References	439
	Security and Software Engineering	445
	Sam Malek, Hamid Bagheri, Joshua Garcia, and Alireza Sadeghi	
1	Introduction	446
2	Concepts and Principles	447
3	Organized Tour: Genealogy and Seminal Works	453
4	Future Challenges	480
5	Conclusions	484
	References	485
	Software Engineering in the Cloud	491
	Eric M. Dashofy	
1	Introduction	491
2	Key Concepts	493
3	The Software Economics of Clouds	505
4	Software Development and Deployment in the Cloud	507
5	Seminal Papers and Genealogy	509
6	Conclusions	512
	References	514
	Index	517

Editors and Contributors

About the Editors

Sungdeok Cha is a Professor at Korea University in Seoul, Korea and a former professor at Korea Advanced Institute of Science and Technology (KAIST) in Daejeon. Prior to joining KAIST, he was a member of technical staff at the Aerospace Corporation and the Hughes Aircraft Company working on various software engineering and computer security projects. His main research topics include software safety, requirements engineering, and computer security. He is also a member of editorial boards for several software engineering journals.

Richard N. Taylor is a Professor Emeritus of Information and Computer Sciences at the University of California, Irvine, USA. His research interests are centered on design and software architectures, especially focusing on decentralized systems. In 2017 he received the ACM SIGSOFT Impact Paper Award (with Roy Fielding). In 2009 he was recognized with the ACM SIGSOFT Outstanding Research Award, in 2008 the ICSE Most Influential Paper award, and in 2005 the ACM SIGSOFT Distinguished Service Award. In 1998 he was named an ACM Fellow for his contributions to research in software engineering and software environments.

Kyochul Kang is an Executive Vice President at Samsung Electronics as well as a Professor Emeritus at POSTECH in Korea. Prior to joining POSTECH, he conducted software engineering research at Bell Communications Research, Bell Labs, and SEI. His research career in software engineering began in the 1970s as a member of the PSL/PSA team, which developed the first-ever requirements modelling and analysis technology. He is well known for his FODA (Feature-Oriented Domain Analysis) work at SEI and is an expert on software reuse and product line engineering.

Contributors

Eduardo Santana de Almeida Federal University of Bahia, Salvador, Bahia, Brazil

Hamid Bagheri University of Nebraska-Lincoln, Lincoln, NE, USA

Amel Bennaceur The Open University, Milton Keynes, UK

Eric M. Dashofy The Aerospace Corporation, El Segundo, CA, USA

Gordon Fraser University of Passau, Passau, Germany

Joshua Garcia University of California, Irvine, CA, USA

Volker Gruhn Lehrstuhl für Software Engineering, Universität Duisburg-Essen, Essen, Germany

Yann-Gaël Guéhéneuc Polytechnique Montréal and Concordia University, Montreal, QC, Canada

Foutse Khomh Polytechnique Montréal, Montreal, QC, Canada

Miryung Kim University of California, Los Angeles, CA, USA

Sam Malek University of California, Irvine, CA, USA

Na Meng Virginia Tech, Blacksburg, VA, USA

Bashar Nuseibeh The Open University, Milton Keynes, UK

Lero The Irish Software Research Centre, Limerick, Ireland

Leon J. Osterweil University of Massachusetts, Amherst, MA, USA

Doron A. Peled Bar Ilan University, Ramat Gan, Israel

José Miguel Rojas University of Leicester, Leicester, UK

Alireza Sadeghi University of California, Irvine, CA, USA

Anita Sarma Oregon State University, Corvallis, OR, USA

Nils Schwenzfeier Universität Duisburg-Essen, Essen, Germany

Tetsuo Tamai The University of Tokyo, Tokyo, Japan

Richard N. Taylor University of California, Irvine, CA, USA

Thein Than Tun The Open University, Milton Keynes, UK

Danny Weyns Katholieke Universiteit Leuven, Leuven, Belgium
Linnaeus University, Växjö, Sweden

Yijun Yu The Open University, Milton Keynes, UK

Tianyi Zhang University of California, Los Angeles, CA, USA