

Action Research in Software Engineering

Mirosław Staron

Action Research in Software Engineering

Theory and Applications

 Springer

Mirosław Staron
Department of Computer Science
and Engineering
University of Gothenburg
Gothenburg, Sweden

ISBN 978-3-030-32609-8 ISBN 978-3-030-32610-4 (eBook)
<https://doi.org/10.1007/978-3-030-32610-4>

© Springer Nature Switzerland AG 2020

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, expressed 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

To my family

Foreword

Action! It is what engineered projects demand. As stated by the Cambridge Dictionary,¹ the action is “*the process of doing something, especially when dealing with a problem or difficulty.*” Usually, the engineering of any product involves a plan of actions. Such a plan must consider the different dimensions of the problem and the context variables (physical, environmental, technical, and social) that somehow influence product construction. However, only effective actions can guarantee the success of projects and the reduction of engineering risks. That is what engineers learned along the centuries.² At least four stages of (r)evolution (pre-scientific, first industrial, second industrial, information) can be observed regarding the understanding of the instruments, technology limits, and properties of different products. The interaction between theory (research) and practice (action) has been vital to acquire evidence to support this learning process. There are indeed plenty of challenges to face in contemporary products and many lessons to learn. In whatever way, the evolution of engineering knowledge allows engineers to offer and build more and more complex solutions to the benefit of society.

The Cambridge Dictionary states that research is “*a detailed study of a subject, especially in order to discover (new) information or reach a (new) understanding.*” The learning process in the engineering field has been supported by research, in its different formats and configurations. In this case, a researcher expects “*to study a subject in detail, especially in order to discover new information or reach a new understanding.*” It is possible to notice the contribution of experimentation strategies in the evolution of engineering by observing the growth of some domains. For instance, some well-established engineering fields, such as automobile, civil,

¹<https://dictionary.cambridge.org/dictionary/english/>.

²<http://www.creatingtechnology.org/history.htm>.

chemistry, or electrical engineering, have identified the primary challenges and context variables that can influence the plan of actions to build their conventional products and that could contribute to risky contemporary ones. The commonality of properties offered by their products (discovered by research) indeed makes the plan of actions less challenging.

Software engineering and its related products are young when compared with other engineering fields. The expected commonality and somehow stable properties presented by conventional engineered products become less tangible when talking about software products. There are many context variables involved in the engineering of a particular software product or software technology. The planning of actions is risky since software products differ from each other in at least one of their planning dimensions (peopleware, processes, and product). Perhaps, part of the challenge comes from the difficulties software engineers face in defining and showing the product they use to build. Besides, there are unknown context variables that can influence the plan of actions (software processes, in this case) and affect the result of the project. Therefore, observation, experimentation, and learning are vital to support the evolution of our capacity of engineering software products and technologies for the benefit of society.

Each software project or software technology development represents an unmissable opportunity to research, learn, and evolve! Software engineers and practitioners indeed learned a lot since the 1950s when software products started to be delivered. However, all of this learning is (and maybe it will never be) not enough to support the building of new products. Software is everywhere. Its costs are concerned with its engineering, on which manufacturing is not possible. It needs to evolve to keep itself valuable and useful, but it deteriorates at the same time. New problems and solutions demand custom-built components, inserting quality risks. Besides, all software systems can fail, introducing damage risks depending on the problem domain. All of these issues can challenge building and maintenance actions. However, they vary according to the features of the different software projects and their context variables. Therefore, they need to be observed, characterized, and mitigated.

As previously demonstrated in other engineering fields, research represents an essential instrument to support the understanding of the software-related phenomena and the mitigation of issues in the software processes. Empirical Software Engineering is an area of software engineering that has intensively worked to understand the application and evolution of software processes and technologies by applying the scientific method (experimentation) and other observation strategies. Nowadays, experimentation is the realm of software engineering when discussing the combination of theory (researchers) and practice (software practitioners) in favor of learning and evolution of the field.

Along the years, the application of software engineering and experimentation principles and strategies in the software projects have been a reality in my professional life. I indeed started doing much more software engineering than

experimentation, next begun to do much more experimentation than engineering software products, and so realized that the merging of actions with research could be great to support development, learning (in general), and the decision-making (in particular) in software projects (mainly those really challenging and unexpected in terms of requirements, technology, innovation, and organization). These experiences were reported in some publications, which intended to make clear (even that not completely) the collaboration involved in those experiences and to share with the software engineering community what we learned and applied in the projects. It was in 2009. The impact of using action and research in our projects was so intense that we decided to call for this strength in the title of one of our publications. Since then, action research became a strategy of engineering in all of the software projects of the Experimental Software Engineering Group at COPPE/UFRJ with the industry that this combination of approaches (action and research) is feasible and makes sense.

It was June 16, 2019, when I received an initial message from Prof. Dr. Mirosław Staron, from the University of Gothenburg, Sweden, talking about the experiences that Paulo Sérgio Medeiros dos Santos and I reported in one of our previous publications in 2011 and the similarities with his current experiences on using action research in the industry. The sequence of messages was full of kindness and included a link to the draft of this book, which I read with great interest and pleasure. It was possible to understand why he sent that initial (and other) messages to me. I felt honored and pleased by receiving the messages, an invitation to prepare this foreword, besides being able to be one of the first readers of his book.

Action Research in Software Engineering: Theory and Applications, by Prof. Mirosław Staron, is a must-read book for those researchers and practitioners interested and concerned with strengthening the collaboration between academia and industry, building a plan of actions based on evidence, or making decisions supported by science in their software projects. It offers 12 chapters full of information and relevant discussions regarding the use and limits of action research in software engineering. The chapters present concrete examples, which make the understanding of concepts easy for those not wholly involved with the empirical software engineering context.

The book covers the cycles of action research. It organizes the chapters in the way an action research strategy is usually introduced into the software projects. The instruments used throughout the cycles of action research can be easily realized or captured from the discussions and examples. This material is of great value for those that need to speed up the introduction of action research and guarantee the effectiveness of actions in their software projects.

This book is also a contribution to the empirical software engineering community. It registers and tailors the processes and principles involved in action research to the software engineering field, describing the methodology and making explicit the limits of action research when applied in the area. Besides, this book also represents an adequate material to support graduate courses regarding action research in software engineering.

Thanks, Miroslaw, for sharing with the software engineers your experiences and knowledge regarding action and research!

I hope the reader enjoys reading this book as much as I did.

Professor of Software Engineering
CNPq Researcher, ISERN Member
Experimental Software Engineering Group
PESC/COPPE/Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
August 2019

Guilherme Horta Travassos

Preface

Every scientist has his and her own favorite research topics, types of studies, and research methodology. It's natural, and it's something that is very important for all kinds of researchers. I see it as a part of academic freedom. For me, the favorite research methodology is action research.

I started with action research without knowing that it is a valid research methodology. A company was looking for a researcher who could help them with research on software metrics. I was interested and met with the company, and so we started. The company asked me to spend all my research project time at their premises, not in my office (which is ca. 300 m away in another building). Colleagues from my department called this a consultancy and not research (at that time), but they supported me.

I'm very grateful for their support, because this kind of working with research turned out to be "my thing," and I've been working according to action research since then. I've learned that the first problem formulation is often symptomatic, the real problem is hidden, and we need to run some diagnostics to find it. At the beginning, this diagnostics took me a lot of time, several interviews, and data collection. In the course of time, it became easier as I learned where to look for. I've also learned how to work with practitioners. Many of my students are now working for the companies that I collaborate with, so we have a common language thanks to the knowledge I got from their older colleagues. I'm very grateful for that.

Today, it's obvious that action research is the methodology that my research team uses. Our industrial partners expect us to work with them, solve their industrial problems, and contribute to theories in software engineering. I wrote this book to help young and experienced researchers, scientists, and software engineers. I would like to inspire them to action and to encourage them to try action research, because it requires courage. As an action researcher, you need to listen to your industrial colleagues; sometimes you need to admit to making a mistake or not knowing how things work in industry.

In this book, I start with the description of action research, its history, and purpose in Chaps. 1 and 2. I provide examples of how a research proposal looks like and why we need to write it in this particular way. In Chap. 3 I go into detail of how

to diagnose the “real” problem that needs to be solved. Chapter 4 is all about the planning of the action, and Chap. 5 is about action taking. Chapter 6 helps with the evaluation and discusses methods for analyzing qualitative and quantitative data. Chapter 7 elaborates on the methods for identifying knowledge which is important for dissemination at the company and in the academic community. Chapter 8 provides an alternative research methodology to action research as a methodology—design science research—and discusses their differences and similarities. Chapter 9 helps to ensure that the knowledge developed in action research lasts longer than one specific project. Chapter 10 is about evaluating the validity of an action research study. It discusses the most common validity threats and how to reduce them. Finally, Chap. 11 describes how to document and report action research studies.

I hope that this book will encourage researchers and practitioners to work together and to use research projects as a means of advancing the field of software engineering.

Gothenburg, Sweden
August 2019

Mirosław Staron

Acknowledgments

This book is a work based on years of experience with industrial research. This experience has been shaped by many collaborators, far too many to thank all of them individually. I am enormously grateful to work with my industrial colleagues.

I would like to thank my colleagues from Ericsson, which is the company where I started my journey with action research. In particular, I would like to thank Wilhelm Meding for the support and encouragement throughout the years, in particular, for the encouragement after my failures. There would be no book if it was not for Wilhelm's encouragement.

I would also like to thank Micael Caiman, who supported me throughout all the years. Micael has always been kind enough to provide me with guidance and ideas, as well as with access to the physical infrastructure at Ericsson. I'm extremely thankful for this, and I hope that Micael knows it.

Ericsson is only one of many companies that supported me. I am indebted to my colleagues at Volvo Cars, who provided me with directions, ideas, and access to their infrastructure, including the possibility to be part of their research efforts. In particular, I would like to thank Kent Niesel, Hans Alminger, Darko Durisic, Martin Nilsson, and Anna Sandberg. They have helped me more than I could have expected and, most probably, more than they realize.

I pitched the idea of this book to some of my colleagues at the department (IT Faculty, Department of Computer Science and Engineering, Software Engineering Division—yes, that is a long name) during a lunch. I got mixed reactions, but some of my colleagues supported and encouraged me, in particular, Agneta Nilsson, Mirosław Ochodek, and Imed Hammouda. Thank you all for this. I really appreciate that. I would also like to thank Richard Berntsson Svensson, who engaged in discussions about the content of the book and pointed me to several publications in this area.

I'm also indebted to my publisher—Ralf Gerstner—who supported me during the process and who bootstrapped my writing process. If you ever need a publisher, I sincerely recommend Ralf. It's difficult to find a more skilled professional.

Finally, I would like to thank my family for the support, encouragement, and kindness during the writing process. The book is an investment of their time, which I hope I can repay one day.

Contents

1	Introduction	1
1.1	Introduction	1
1.2	What Action Research Is	2
1.3	The Concept of Action	3
1.4	Short Historical Timeline of Action Research	4
1.5	Software Engineering and Its Context	4
1.6	Action Research in Software Engineering	6
1.7	So, What We Need for Action Research Is... ..	6
1.8	Outline of This Book	7
1.8.1	Chapter 2: Action Research as Research Methodology in Software Engineering	8
1.8.2	Chapter 3: Diagnosing	8
1.8.3	Chapter 4: Action Planning	8
1.8.4	Chapter 5: Action Taking	9
1.8.5	Chapter 6: Evaluating	9
1.8.6	Chapter 7: Learning	9
1.8.7	Chapter 8: Action Research vs. Design Science Research	10
1.8.8	Chapter 9: Ensuring Sustainability of Knowledge	10
1.8.9	Chapter 10: Validity Evaluation of Action Research Studies	11
1.8.10	Chapter 11: Reporting Action Research Studies	11
1.9	Let's Go!	11
	References	12
2	Action Research as Research Methodology in Software Engineering	15
2.1	Introduction	15
2.2	Phases of Action Research Cycles	17
2.2.1	Diagnosing	18
2.2.2	Action Planning	18

- 2.2.3 Action Taking 19
- 2.2.4 Evaluation 19
- 2.2.5 Learning 20
- 2.3 Action Research Cycles in Software Engineering 20
 - Organizations 20
- 2.4 Actors in Action Research 22
- 2.5 Number of Cycles 22
- 2.6 Collecting the Data 23
- 2.7 Visualizing the Data 23
- 2.8 Software Experiment Systems 25
- 2.9 An Action Research Project Proposal 28
 - 2.9.1 Context 30
 - 2.9.2 Goal 31
 - 2.9.3 Actors 31
 - 2.9.4 Intended Outcome 32
 - 2.9.5 Actions 32
 - 2.9.6 Data Collection and Measurements 32
 - 2.9.7 Evaluation 32
 - 2.9.8 What We Want to Learn 33
 - 2.9.9 Ethical Considerations 33
- 2.10 Managing Action Research Projects 34
- 2.11 Summary and Conclusions 35
- References 35
- 3 Diagnosing 37**
 - 3.1 Introduction 37
 - 3.2 Role of Theory in Diagnosing 38
 - 3.3 Interviews 40
 - 3.3.1 Preparing for Interviews 41
 - 3.3.2 Documenting Interviews 42
 - 3.3.3 Appreciative Inquiry 43
 - 3.4 Observations 43
 - 3.4.1 Preparing for Observations 44
 - 3.4.2 Documenting Observations 45
 - 3.4.3 Participatory Observations 45
 - 3.5 Focus Group Workshops 46
 - 3.5.1 Preparing for Focus Group Workshops 48
 - 3.5.2 Documenting Focus Group Workshops 50
 - 3.6 Collecting Quantitative Data 50
 - 3.7 What Each Role Does in This Phase 51
 - 3.8 Example of Results from the Diagnosis Phase 52
 - 3.9 Formulating Research Questions 52
 - 3.10 Diagnosing Phase in Second, Third, and the Subsequent
Cycles 54
 - 3.11 Software Experiment Systems and Diagnosing 55

- 3.12 Summary 55
- References 56
- 4 Action Planning** 57
 - 4.1 Introduction 57
 - 4.2 Access to Competence 59
 - 4.3 Access to Infrastructure 60
 - 4.4 Planning of Actions, Activities, and Participants 61
 - 4.5 Planning of Milestones 62
 - 4.6 Planning of Deliverables 64
 - 4.7 Planning of Status/Planning Meetings 65
 - 4.7.1 Planning of Presentations to the Stakeholders 66
 - 4.8 Planning of Writing Up the Results 67
 - 4.9 Planning in the Second and Subsequent Cycles 67
 - 4.10 Roles in Action Planning and the Process of Creating the Plan 68
 - 4.11 Example of an Action Plan 68
 - 4.12 Action Planning in Experiment Systems 70
 - 4.13 Summary 71
 - References 72
- 5 Action Taking** 73
 - 5.1 Introduction 73
 - 5.2 Actions in Action Research 76
 - 5.3 Principles of Action Taking: Similarity to Experiment Trials 77
 - 5.4 Before Taking the Action 80
 - 5.4.1 Preparing the Evaluation Framework 80
 - 5.4.2 Preparing the Access to the Infrastructure 81
 - 5.5 Taking the Action 84
 - 5.5.1 Individual Actions 85
 - 5.5.2 Actions Involving Others, Like Teams
of Organizations 86
 - 5.5.3 Actions Involving Customers 87
 - 5.5.4 Knowing When to Pivot 87
 - 5.6 Collecting the Data 87
 - 5.7 Using Experiment Systems to Collect the Data: A Special
Case of Action Taking 90
 - 5.8 Summary 91
 - References 91
- 6 Evaluation** 93
 - 6.1 Introduction 93
 - 6.2 Cleaning and Preparing the Data 95
 - 6.3 Data Visualization 97
 - 6.4 Descriptive Statistics 98
 - 6.5 Basics of Inferential Statistics 100
 - 6.6 Machine Learning Methods 102

- 6.6.1 Clustering 103
- 6.6.2 Classification..... 107
- 6.7 Analysis of Qualitative Data 112
 - 6.7.1 Becoming Familiar with the Data..... 112
 - 6.7.2 Generating Initial Codes 113
 - 6.7.3 Searching for Themes 115
 - 6.7.4 Reviewing Themes 116
 - 6.7.5 Defining and Naming Themes 118
- 6.8 Continuous Data Analysis 118
- 6.9 Evaluation in the Second and Subsequent Action
Research Cycles 120
- 6.10 Evaluation in Software Experiment Systems 120
- 6.11 Summary..... 120
- References..... 121
- 7 Specifying Learning 123**
 - 7.1 Introduction..... 123
 - 7.2 Specifying Learning for Companies and Teams 124
 - 7.2.1 Active Learning..... 132
 - 7.2.2 Template for Specifying Learning 132
 - 7.3 Contributing to Theory Building 133
 - 7.3.1 Examples of Types of Contributions..... 134
 - 7.4 Specifying Learning from Experiment Systems 138
 - 7.5 Summary..... 138
 - References..... 139
- 8 Action Research vs. Design Research 141**
 - 8.1 Introduction..... 141
 - 8.2 Design Research..... 142
 - 8.2.1 Awareness of the Problem or Problem Investigation..... 143
 - 8.2.2 Treatment or Artifact Design..... 144
 - 8.2.3 Treatment or Artifact Implementation 144
 - 8.2.4 Treatment or Artifact Evaluation 144
 - 8.3 Similarities and Differences 144
 - 8.3.1 When to Choose Which Methodology..... 148
 - 8.4 Summary..... 150
 - References..... 150
- 9 Ensuring Sustainability of Knowledge 153**
 - 9.1 Introduction..... 153
 - 9.2 Researcher’s Perspective on the Sustainability of Knowledge 154
 - 9.3 Practitioner’s Perspective on the Sustainability of Knowledge 156
 - 9.4 Diagnosing..... 156
 - 9.5 Action Planning 157
 - 9.6 Action Taking 160
 - 9.7 Evaluating 162

- 9.8 Specifying Learning..... 165
- 9.9 Summary..... 165
- References..... 167
- 10 Validity Evaluation 169**
 - 10.1 Introduction..... 169
 - 10.2 Construct Validity 170
 - 10.3 Internal Validity 175
 - 10.4 Conclusion Validity 180
 - 10.5 External Validity..... 185
 - 10.6 Summary..... 188
 - References..... 188
- 11 Reporting Action Research Studies..... 191**
 - 11.1 Introduction..... 191
 - 11.2 Basics of Reporting Action Research Studies 192
 - 11.3 Working with Results That Are Sensitive to Our Industrial Partners 197
 - 11.4 Reporting Studies Focused on Results 200
 - 11.5 Reporting Studies Focused on Actions 206
 - 11.6 Summary..... 211
 - References..... 212
- 12 Conclusions..... 215**
 - 12.1 Experiences from Working According to Action Research 215
 - 12.2 Where Action Research Fits Best 216
 - 12.3 Combining Action Research with Other Methodologies 217
 - 12.4 Where to Go Next: Action Research with Multiple Companies 218
 - 12.5 Final Remarks 220
 - References..... 220