

Rashina Hoda (Ed.)

LNBIP 364

Agile Processes in Software Engineering and Extreme Programming – Workshops

XP 2019 Workshops
Montréal, QC, Canada, May 21–25, 2019
Proceedings

Lecture Notes in Business Information Processing

364

Series Editors

Wil van der Aalst

RWTH Aachen University, Aachen, Germany

John Mylopoulos

University of Trento, Trento, Italy

Michael Rosemann

Queensland University of Technology, Brisbane, QLD, Australia

Michael J. Shaw

University of Illinois, Urbana-Champaign, IL, USA

Clemens Szyperski

Microsoft Research, Redmond, WA, USA


More information about this series at <http://www.springer.com/series/7911>

Rashina Hoda (Ed.)

Agile Processes in Software Engineering and Extreme Programming – Workshops

XP 2019 Workshops
Montréal, QC, Canada, May 21–25, 2019
Proceedings

Editor

Rashina Hoda 
University of Auckland
Auckland, New Zealand



ISSN 1865-1348 ISSN 1865-1356 (electronic)
Lecture Notes in Business Information Processing
ISBN 978-3-030-30125-5 ISBN 978-3-030-30126-2 (eBook)
<https://doi.org/10.1007/978-3-030-30126-2>

© The Editor(s) (if applicable) and The Author(s) 2019. This book is an open access publication.

Open Access This book is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this book are included in the book's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the book's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

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

Preface

This volume contains the research workshops and doctoral symposium, as well as panel summaries presented at XP 2019, the 20th International Conference on Agile Software Development, held during May 21–25, 2019, at the École de technologie supérieure in Montréal, Québec, Canada.

XP is the premier agile software development conference combining research and practice. Whether you were new to agile or a seasoned agile practitioner, XP 2019 provided an informal environment to network, share, and discover trends in agile for the next 20 years.

Research papers and talks submissions were invited for the three XP 2019 research workshops, namely, Agile Transformation, Autonomous Teams, and Large Scale Agile. The workshops attracted a variety of agile software development topics related to the respective workshops. These included papers and presentations based on empirical studies and those conducted in industrial settings relevant to both researchers and practitioners. Submissions in various stages of progress were encouraged to generate discussions at the workshops. The workshops were structured to include a combination of talks, including keynotes by researchers leading the field in the respective areas, a variety of research talks by new, emerging, and established researchers, and round-table discussion style sessions. Research agendas for the communities represented by the workshops were discussed, refined, and reported in the respective workshop summaries.

The research workshops and doctoral symposium provide a highly relevant, friendly, and interactive platform to share and discuss emerging and late breaking research findings. They represent smaller, close communities of passionate emerging and established researchers and a psychologically safe environment to provide and receive feedback.

The success of the XP 2019 research workshops and doctoral symposium was possible due to the contributions of the organizers, Program Committee members, authors, presenters, and general attendees. Last, but not least, I would like to express my sincere thanks to the XP conference Steering Committee and the Agile Alliance for their ongoing support.

Panels at XP 2019 have been consistently among the best attended and well-received conference sessions following the conference plenary keynotes. This year's topics included: Security and Privacy; The Impact of the Agile Manifesto on Culture, Education, and Software Practices; Business Agility – Agile's Next Frontier; and "Agile – The Next 20 Years." In the past we have published panel abstracts with panellist bios and position statements prior to the conference – however these abstracts did not reflect what was discussed at the conference.

For XP 2019, we took a different approach. We prepared extended summaries of the panel discussions, which captured the panellists' initial reflections on the panel topic, the principal questions from the floor, and panellist discussion. These summary articles

were forwarded to the panellists for review, and the reviewed summaries have been included in these post-conference proceedings. We hope that you find this content useful and relevant to both software research and industrial practice.

Special thanks go to Dennis Mancl for his diligence in capturing and editing the panel conversations, the panellists for their participation and content review, and to the XP 2019 panel audiences for their questions and comments.

May 2019

Rashina Hoda
Steven Fraser

Organization

Conference Chair

François Coallier École de technologie supérieure, Canada

Research Workshops Chair

Rashina Hoda The University of Auckland, New Zealand

Agile Transformation Workshop Co-organizers

Leonor Barroca The Open University, UK
Torgeir Dingsøy SINTEF, Norway
Marius Mikalsen SINTEF, Norway

Autonomous Teams Workshop Co-organizers

Nils Brede Moe SINTEF, Norway
Viktoria Stray University of Oslo, SINTEF, Norway

Large Scale Agile Workshop Organizer

Julian Bass University of Salford, UK

Educator Symposium Co-chairs

Craig Anslow Victoria University of Wellington, New Zealand
Martin Kropp Fachhochschule Nordwestschweiz, Switzerland

Panels Chair

Steven Fraser Innovec Innovation, USA

Doctoral Symposium Co-chairs

Casper Lassenius Aalto University, Finland
John Noll University of East London, UK

Publication Chair

Philippe Kruchten The University of British Columbia, Canada

Agile Transformation Workshop Program Committee

Finn Olav Bjørnson	Norwegian University of Science and Technology, Norway
Peggy Gregory	University of Central Lancashire, UK
Tomas Gustavsson	Karlstad University, Sweden
Parastoo Mohaghegi	Norwegian Welfare Administration, Norway
Marius Mikalsen	SINTEF Digital, Norway
Teemu Karvonen	University of Oulu, Finland
Casper Lassenius	Alto University, Finland Simula Research Laboratory, Norway
Helen Sharp	Open University, UK
Katie Taylor	Agile Business Consortium, UK

Autonomous Teams Workshop Program Committee

Antonio Martini	University of Oslo, Norway
Bjørnar Tessem	University of Bergen, Norway
Dag Sjøberg	University of Oslo, Norway
Darja Šmite	Blekinge Institute of Technology, Sweden
Gunnar Bergersen	University of Oslo, Norway
Jutta Eckstein	IT Communication, Germany
Maria Paasivaara	IT-University of Copenhagen, Denmark
Parastoo Mohaghegi	Norwegian Labour and Welfare Administration, Norway
Marius Mikalsen	SINTEF, Norway
Rini Van Solingen	Delft University of Technology, The Netherlands
Thomas Gustavsson	Karlstads Business School, Sweden

Large-Scale Agile Workshop Program Committee

Steve Adolph	cPrime, Canada
Finn Olav Bjørnson	NTNU, Norway
Torgeir Dingsøy	SINTEF, Norway
Jutta Eckstein	IT Communication, Germany
Peggy Gregory	UCLAN, UK
Tomas Gustavsson	Karlstad University, Sweden
Andy Haxby	Competa, The Netherlands
Aymeric Hemon	Université de Nantes, France
Eric Knauss	Gothenburg University, Sweden
Maarit Laanti	Nitor Delta, Finland
Carl Marnewick	University of Johannesburg, South Africa
Nils Brede Moe	SINTEF, Norway
Helena Holmstrom Olsson	Malmö University, Sweden
Maria Paasivaara	Aalto University, Finland
Alexander Poth	Volkswagen, Germany

Ken Power
Klaas-Jan Stol

Cisco, Ireland
Lero, Ireland

Steering Committee

Juan Garbajosa (Chair)	Universidad Politécnica de Madrid, Spain
Ademar Aguiar	Universidade do Porto, Portugal
Hubert Baumeister	Technical University of Denmark, Denmark
Philip Brock	Agile Alliance, USA
François Coallier	École de technologie supérieure, Canada
Jutta Eckstein	Independent, Germany
Steven Fraser	Innoxec, USA
Casper Lassenius	Aalto University, Finland
Erik Lundh	Ingenjörsglädje, Sweden
Michele Marchesi	University of Cagliari, Italy
Maria Paasivaara	Aalto University, Finland
Seb Rose	Cucumber Limited, UK
Viktoria Stray	University of Oslo, SINTEF, Norway
Nils Wloka	Codecentric, Germany

Contents

First International Workshop on Agile Transformation

Agile Transformation: A Summary and Research Agenda from the First International Workshop	3
<i>Leonor Barroca, Torgeir Dingsøy, and Marius Mikalsen</i>	

Second International Workshop on Autonomous Teams

Trends and Updated Research Agenda for Autonomous Agile Teams: A Summary of the Second International Workshop at XP2019	13
<i>Nils Brede Moe, Viktoria Stray, and Rashina Hoda</i>	

Spotify Tailoring for Promoting Effectiveness in Cross-Functional Autonomous Squads	20
<i>Abdallah Salameh and Julian M. Bass</i>	

Voices from the Teams - Impacts on Autonomy in Large-Scale Agile Software Development Settings.	29
<i>Tomas Gustavsson</i>	

Exploring the Challenges of Integrating Data Science Roles in Agile Autonomous Teams.	37
<i>Ivar Hukkelberg and Marthe Berntzen</i>	

The Influence of Culture and Structure on Autonomous Teams in Established Companies.	46
<i>Simone V. Spiegler, Christoph Heinecke, and Stefan Wagner</i>	

Agile Autonomous Teams in Complex Organizations	55
<i>Marius Mikalsen, Magne Næsje, Erik André Reime, and Anniken Solem</i>	

Earn Your Wings: A Novel Approach to Deployment Governance	64
<i>Yvan Petit and Carl Marnewick</i>	

7th International Workshop on Large-Scale Agile

Future Trends in Agile at Scale: A Summary of the 7 th International Workshop on Large-Scale Agile Development	75
<i>Julian M. Bass</i>	

SAFe Adoptions in Finland: A Survey Research.	81
<i>Maarit Laanti and Petri Kettunen</i>	

Comparing Scaling Agile Frameworks Based on Underlying Practices 88
Sven Theobald, Anna Schmitt, and Philipp Diebold

Finnish Enterprise Agile Transformations: A Survey Study 97
Petri Kettunen, Maarit Laanti, Fabian Fagerholm, Tommi Mikkonen, and Tomi Männistö

Changes Over Time in a Planned Inter-team Coordination Routine 105
Tomas Gustavsson

Technical-, Social- and Process Debt in Large-Scale Agile:
An Exploratory Case-Study 112
Antonio Martini, Viktoria Stray, and Nils Brede Moe

Doctoral Symposium

Coordination in Large-Scale Agile Software Development 123
Marthe Berntzen

Panels

XP 2019 Panel: Security and Privacy 137
Dennis Mancl and Steven D. Fraser

XP 2019 Panel: Agile Manifesto – Impacts on Culture, Education,
and Software Practices 143
Dennis Mancl and Steven D. Fraser

XP 2019 Panel: Business Agility 149
Dennis Mancl and Steven D. Fraser

XP 2019 Panel: Agile, the Next 20 Years 154
Dennis Mancl and Steven D. Fraser

Author Index 159



Earn Your Wings: A Novel Approach to Deployment Governance

Yvan Petit¹(✉) and Carl Marnewick²

¹ Business School of University of Quebec at Montreal, Montreal, Canada
petit.yvan@uqam.ca

² University of Johannesburg, Johannesburg, South Africa

Abstract. This paper presents a model to assess team autonomy developed and deployed in a South African bank's IT department. The bank has been deploying SAFe® for the last two years and in the process has increased significantly the number of software releases. Historically, the teams had to obtain multiple levels of authorization prior to a release but this level of governance and control was contradictory to the principle of team empowerment at the core of agile approaches. The model is inspired from the theme of a pilot's ability to fly an aircraft using five levels. The level is determined based on team fly-ability and elevation safety described in detail in this paper. Team fly-ability includes two elements: (1) maturity of engineering practices and (2) the ability to manage traceability and risk through ease of recovery. Elevation Safety is based on two components: (1) historical data on deployment performance and severity of incidents and (2) the application dynamics and criticality. The main benefits of this program are improved accountability of teams, reduced approval time, and reduced attempts to find workarounds and loopholes.

Keywords: Scaling agile · Team autonomy · Deployment · Governance

1 Introduction

More and more large organizations are now adopting agile enterprise-wide. However, the scaling of agile to a larger undertaking, creates an entire new set of problems and challenges ranging from resistance to change to the problems associated with hierarchical management and organizational boundaries. Leffingwell [1] groups the challenges of scaling agile into two broad categories: (1) those inherent to the methods themselves, "because of the fixed rule bases and assumptions built into the methods" (p. 87); and (2) challenges imposed by the enterprise that "will prevent the successful application of the new methods" (p. 87). More recently, Dikert, Paasivaara and Lassenius [2] surveyed 52 publications describing 42 industrial cases and reported 35 challenges grouped in nine categories: change resistance, lack of investment, agile difficult to implement, coordination challenged in multi-team environment, different approaches emerge in a multi-team environment, hierarchical management and organizational boundaries, requirement engineering challenges, quality assurance challenges, integrating non-development functions. Surprisingly, very little empirical research has been done on how to alleviate these challenges [3].

However, consultancy firms have developed various frameworks and models to address these challenges to implement scaled agile: Disciplined Agile Framework [4], Large Scale Scrum (LeSS) [5] and Nexus [6]. The most popular framework for scaling agile remains SAFe [6–8].

All these frameworks are building on the values and principles of the agile manifesto [9], among others the deployment and support of self-organized teams. Although much has been researched on teams, particularly in the field of psychology, much remains to be investigated in the context of agile deployment. The objective of this paper is therefore to contribute to the field, based on a research question proposed, during last year’s workshop on scaling agile at XP 2018 [10]: “*What is the right degree of team autonomy in different contexts (and how to measure it)?*”

2 Literature Review

Parker, Holesgrove, and Pathak [11] define a self-organized team “as a self-regulated, semi-autonomous small group of employees whose members determine, plan and manage their day-to-day activities and duties under reduced or no supervision.” (p. 112) and the labels “autonomous teams” have been used as synonyms for “self-organizing teams,” “self-managing teams” and “empowered teams” [10].

Teams have been studied for decades, primarily in the field of psychology. Literature abounds on numerous topics such as: leadership, roles, phases, performance, team dynamics, etc. More and more researchers reuse some of that literature to study software development teams using agile approaches, for example to relate autonomy to team performance [12], empowerment and outcomes in software development organizations [13], productivity of self-organized teams [11] and job satisfaction and/or motivation [14].

However, according to Kakar [15] “no approach or instrument has yet been designed to measure and compare self-organization between teams. Second, although conceptually the difference in adoption of self-organization in agile versus plan-driven methods has been discussed previously in the literature, the levels of self-organization between these two major paradigms of software development have not been objectively compared” (p. 208). In the case presented in this paper, the issue was the assessment of the teams and the level of authority delegated to them by the senior management. This is what Cao, Mohan, Xu and Ramesh classified as “sources of structure” in their proposed framework.

3 Methodology

The development and deployment of a new model to assess teams in order to grant them the right level of autonomy to release software was observed in a large South African bank’s IT department. Twenty employees in the department were interviewed. The interviews were conducted by the two researchers themselves. The twenty interviewees comprised of four people from business who were direct customers of IT and ultimately the agile process, two people from the agile portfolio office, four portfolio

project managers, the CFO of the IT department, two CIOs within the IT department, three release train engineers, one COO and three agile coaches.

Most of the interviews were directed towards understanding the link between the IT initiatives and the organization's strategy. However, during the course of the research many interviewees kept referring to an internal program to assess a team's autonomy with respect to the deployment of software. There were two interviews with the people responsible for the deployment and improvement of engineering practices. The interviews were coded in ATLAS.ti to assist in the analysis and summary of the approach. The objective of this paper is primarily to present and share an experience report from the perspective of the bank with the understanding that their approach might be of interest to other organizations.

4 Case Description

SA Bank (Note: This is a fictitious name to protect the identity of the bank), a financial institution, is one of the largest African financial institutions by assets. SA Bank is among the largest organizations in South Africa by market capitalization. It offers a wide range of banking and financial services to millions of personal customers in 20 African countries. The company employs over 6000 people in their IT.

SA Bank decided to start deploying agile practices in their IT department. They redesigned their software development completely and restructured around self-organizing teams responsible for products rather than specific activities. These teams would also be in close contact with business representatives and would be entirely responsible for the development, testing, deployment, and maintenance of their products. The intention was to reduce the number of handoffs and simplify the development process dramatically. The SAFe deployment was strongly committed to and supported by executives and line managers across the board.

At the time of the interviews, SA Bank had deployed SAFe 4.5 but used only three of the four layers [8, 16]: They did not use the "large solution" layer but had the intention to deploy it in coming months.

Semi-permanent self-organizing agile development teams were put in place to support the planning, prioritization, harmonization, synchronization, development and release of the features/systems. The term "semi-permanent" is used to refer to teams being maintained as permanent as possible with the exception of changes related to turnover, punctual member swaps due to competence requirements, etc. The agile teams use many of the Scrum tools and artefacts such as backlogs, burn-down charts and Kanban walls.

5 Background Leading to the Introduction of Earn Your Wings

An incident in March 2017 triggered the introduction of the "Earn Your Wings" program. The teams were used to systematically obtain authorization to release software from a committee, composed of senior managers, called the *change advisory*

board (CAB). One of the teams wanted to deploy a change into production. The changes would have been deployed using a fully automated pipeline and the team was convinced that the software would not be faulted. That release happened to occur in a high-care period. This meant that the team required authorizations from the production manager, from the business responsible, then from the area senior management head sign-off and then finally a sign-off from the CAB.

One of the people responsible for process improvement and SAFe deployment started to challenge the governance and approval process. How could they be talking about team autonomy and team self-organization while at the same time imposing a very cumbersome approval process?

After a number of focus groups, interviews and surveys, they realized that this old approval process actually had three negative side effects (what they called reverse incentive behaviors):

1. The CAB was shielding people from consequence management. If a team was deploying a change into production, so long as they received CAB sign-off, they felt relieved from taking responsibility if anything went wrong.
2. The second reverse incentive was low-low changes. Only low impact and low risk changes were self-governed. If changes were categorized as low-low, the teams did not have to go to CAB. Consequently, more and more teams were categorizing everything as low-low with the risk of deploying faulty high-risk/high-impact packages.
3. If the approval process is tedious, teams tend to package the release in larger bundles and decrease the release frequency

6 Earn Your Wings

In order to reverse these negative side effects, the team in charge of improving the deployment process at SA Bank derived a detailed evaluating and rating scheme. This section summarizes what the bank representatives presented to the researchers. They used the theme of a pilot's ability to fly an aircraft using five levels, from lowest to highest:

- **Level 1 Red Bull:** This is the lowest level for which the highest level of authorization is required. This comes from the image that "Red Bull gives you wings" but actually you are not really able to fly. In other words, the team (and the supported software) is at the lowest level of autonomy.
- **Level 2 Hot Air Balloon:** Although you can get into the air, you are reliant on fire, which can easily go up in flames. You are restricted by conditions as to whether it can take off or not. You carry very little safety gear.
- **Level 3 Small helicopter:** You have freedom of movement but with extra caution. Individually you decide if this is trusted and if the risk is worth taking. Although you can get somewhere, with relative ease, you do not have the best safety gear if something goes wrong

- **Level 4 Dinky Plane:** While you have the freedom to go wherever you want, there are situations in which you would choose not to fly. A trusted form of transport, with a little extra convincing
- **Level 5 Private Plane (highest):** Freedom to go wherever you want, whenever you want. The most trusted form of transport. Air traffic control simply coordinates but does not question.

The level is determined based on **Team Fly-ability** and **Elevation Safety** described in detail in this section.

6.1 Team Fly-Ability

Team fly-ability includes two elements “Maturity of engineering practices” (60%) and “Ability to manage traceability and risk through ease of recovery” (40%).

Maturity of Engineering Practice

The teams subjectively assess their own level of Maturity of Engineering Practices based on: the level of automation, the level of testing, the size of work package and an assessment to whether acceptance criteria are met. In practice, this is performed through a tool called *Continuum* using elements such as coding practices, continuous integration, incident management, release management, quality assurance and risk management. If teams decide to overrate themselves, they take on the risk of something going wrong.

Ease of Recovery

The ease of recovery evaluates the ability to manage traceability and risk, the time to recover and the ability to roll back and restore. This is assessed, using a tool called *Remedy*, based on the Mean Time to Recover. The objective is to maintain the down time under the agreed levels contracted in the Service Level Agreements.

6.2 Elevation Safety

Elevation Safety is based 100% on historic data on deployment performance and severity of incidents with some consideration of the application dynamics and criticality being taken into account.

Historic Data on Deployment Performance and Severity of Incidents

This is based on the performance of the previous five deployments answering questions such as: Have you caused incidents within the two weeks after your deployment due to the change? Have you had an avoidable production incident? What was the frequency of deployments?

Application Dynamics and Criticality

This is based on the chief information officer (CIO)’s list of the most critical applications with respect to: number of users, rate of change, number of dependent systems, the number of countries impacted.

6.3 Overall Assessment on a Scale of Five Levels

Using the combination of team fly-ability and elevation safety described in the previous sections, the overall level of autonomy (from Red Bull to Private Plane) is assessed using the grid in Fig. 1.

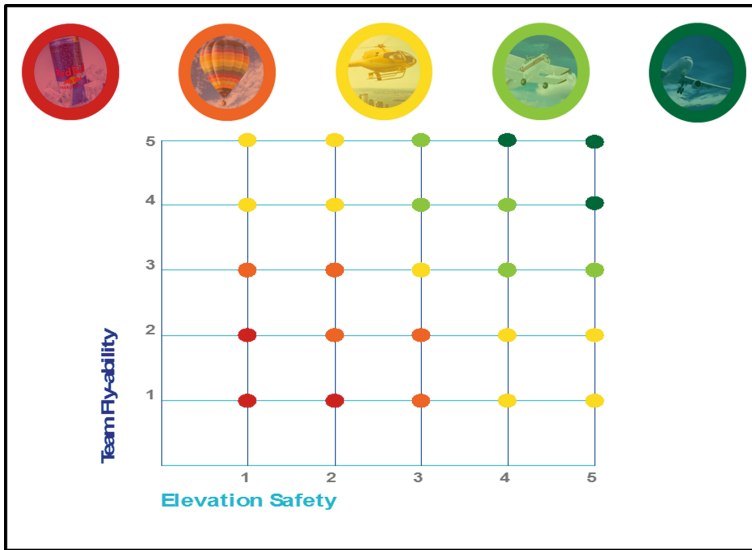


Fig. 1. Overall assessment of the team autonomy

The wings are assigned to teams, not individuals. The teams are continually assessed using real-time data (for example on the quality of their prior releases).

6.4 Authorizations Required to Deploy

The release and deployment authorizations required, depends on the level of the team as follows. This varies from required authorization of the CAB for all changes, for Level 1 (Red Bull), to complete autonomy for releasing and deployment at Level 5 (Private plane). In the case of level 5, the CAB would just be informed and would not be involved in the decision. There are various degrees of authorizations required for intermediate team level medium or above. Depending on the level, the release and deployment constraints are specified e.g. ability to release during the freeze periods, time of the day when deployment is allowed, frequency of deployment (per week), artefacts required.

7 Conclusion

At the time of the interviews, 70 teams had “earned their wings.” The objective was to deploy to approximately 300 teams. Consequently, the organization was working on the improvement of engineering practices to support teams. They also continued to automate data collection to support team evaluation. As far as the SA Bank was concerned, the main benefits of this program had been: (1) Improved accountability of teams (2) reduced duration of the CAB (3) reduced attempts to find workarounds and loopholes and (4) coordination done at the right level i.e. teams govern each other as opposed to management doing it. Release management does not question the releases; they just make sure that there is no mid-air collisions (like air traffic controllers).

Although this research is based on a limited set of interviews, the authors wanted to share how SA bank had used an objective assessment of the autonomy and maturity of the teams as a good example of how an organization tried to answer the research question, “*What is the right degree of team autonomy in different contexts (and how to measure it)?*” Their objective, in doing so, was to improve the deployment process both in terms of quality and the time required to approve the releases.

References

1. Leffingwell, D.: *Scaling Software Agility: Best Practices for Large Enterprises*. Pearson Education, New York (2007)
2. Dikert, K., Paasivaara, M., Lassenius, C.: Challenges and success factors for large-scale agile transformations: a systematic literature review. *J. Syst. Softw.* **119**, 87–108 (2016)
3. Hobbs, B., Petit, Y.: Agile methods on large projects in large organizations. *Proj. Manag. J.* **48**(3), 3–19 (2017)
4. Ambler, S., Lines, M.: *Disciplined Agile Delivery. An agile process decision framework for the enterprise* (2014). <http://disciplinedagiledelivery.com/>
5. Vaidya, A.: Does dad know best, is it better to do less or just be safe? Adapting scaling agile practices into the enterprise. In: *Pacific Northwest Software Quality Conference (PNSQC)*, Portland, OR (2014)
6. Mashal, A., Rozilawati, R.: A review of scaling agile methods in large software development. *Int. J. Adv. Sci.* **6**(6), 828–837 (2016)
7. Scaled Agile, Inc. and VersionOne present webinar on the indispensable elements of SAFe, Amman (2016)
8. Leffingwell, D.: *SAFe® 40 Reference Guide: Scaled Agile Framework® for Lean Software and Systems Engineering*, p. 576. Addison-Wesley Professional, Boston (2017)
9. Agile Alliance: *Manifesto for Agile Software Development* (2001). <http://agilemanifesto.org/>. Accessed 15 Jan 2013
10. Stray, V., Moe, N.B., Hoda, R.: Autonomous agile teams: challenges and future directions for research. In: *Proceedings of the 19th International Conference on Agile Software Development Companion (XP 2018)*. ACM, New York (2018)
11. Parker, D.W., Holesgrove, M., Pathak, R.: Improving productivity with self-organised teams and agile leadership. *Int. J. Prod. Perform. Manag.* **64**(1), 112–128 (2015)
12. Cordery, J.L., et al.: The impact of autonomy and task uncertainty on team performance: a longitudinal field study. *J. Organ. Behav.* **31**(2/3), 240–258 (2010)

13. Tessem, B.: The customer effect in agile system development projects. A process tracing case study. *Proc. Comput. Sci.* **121**, 244–251 (2017)
14. Tripp, J.F., Riemenschneider, C., Thatcher, J.B.: Job satisfaction in agile development teams: agile development as work redesign. *J. Assoc. Inf. Syst.* **17**(4), 267–307 (2016)
15. Kakar, A.K.: Enhancing reflexivity in software development teams: should we focus on autonomy or interdependence? *J. Inf. Technol. Theory Appl.* **17**(3), 5–23 (2016)
16. Leffingwell, D.: SAFe - scaled agile framework (2015). <http://www.scaledagileframework.com/>. Accessed 25 Mar 2019

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

